Crawling with BFS and Random Walk

Social Network Analysis for Computer Scientists — Course Project Paper

Athanasios Agrafiotis
LIACS, Leiden University
a.agrafiotis@umail.leidenuniv.nl

ABSTRACT

In this research, we focused on the collection of data from online social networks. We simulate how a web crawler can collect massive data using two different sampling methods. Until now further researches have been done on datasets although the data collection process plays a crucial role and can determine how biased is our dataset and how close to the ground truth. To evaluate the sample we consider several factors such as the protected users, the behaviour of our sampling algorithms,the structure of the network and the size of the sample.

1. INTRODUCTION

The last decade Online Social Networks (OSNs) growth has drawn the attention for analysis and research. Sociologists and computer scientists are interested in analyzing and researching them. As an OSN we consider every platform which a user can interact with other users such as Youtube and Facebook, these interactions give us the opportunity to understand users preferences and to gain knowledge about their behaviour. However, in order to have accurate results and understand a graph we need the complete data of it.
The big amount of users have contributed to the creation of large graphs which are unfeasible to store it and to calculate their characteristics. It's not a trivial problem to find a way to collect data from OSN like Twitter, Facebook, LinkedIn we can simply use APIs or web crawlers. To make things even worst APIs which suppose to help crawl data have limited the time and request. An efficient representative sample of the network would be the solution to this problem. But how can you find a representative sample ?
To address this challenge sampling methods have developed which by sampling users of an OSN they can give us a representative sample of the whole network. Sufficient sampling methods are those which can provide to us a simple and inexpensive solution.
A lot of research have been done about the sampling goals and sampling algorithms. Different scenarios and different sampling goals have been study in depth. How we can crawl a large graph [4] and how can we retain the graph properties of the graph by sampling with different algorithms [2]. Before we proceed we must also consider that our research is to simulate a process of a web crawler . So our sampling scenario is that we have *restricted access* to the network. The network is hidden, however it supports crawling . That means that it allows us to explore the neighbor of a given user or node . We must also mention that there are still two scenarios *full access* and *streaming access* .
The sampling methods that we can use to crawl a social network are very important can be classified into two categories: a) *Graph traversal techniques* and b) *Random walks.* We seen in depth those techniques later.After the process of crawling, we analyze the data that we have gather .
We will focus how sampling algorithms can crawl a large social network and what sample can give us. We consider the links between users as edges and the users as nodes. We will evaluate the efficiency of the sampling algorithms according to the size of the sample and properties which retain identical of our large graph. We will use the Breadth First Search (BFS) algorithm and the Metropolis Hasting Random Walk (MHRW) algorithm. We will explain how these algorithms work later.
In the second phase, we evaluate the above algorithms if they maintain properties of the original graph. These properties are the node degree distribution, (NDD) and the average degree. (AC) [2, 3, 4]; . And we will compare these properties with the *ground truth.*
Our research is different from others, because we will try to simulate the way that a web crawler collects data from a social network graph using two different crawling techniques by collecting the same portion of nodes and links. And we will evaluate the sample that we gain.

The rest of the paper is organized as follows. In Section II we specify the major problems of crawling. In Section III we introduce some previous work and how it related to our work. In Section IV sampling algorithms how they work the two sampling algorithms that we will use for our experiment and which are the factors that we need to consider; we make some assumptions and which graph properties we will use. In Section V is our datasets and the evaluation of our experiment and finally in Section VI our conclusion and future work.

2. PROBLEM STATEMENT

As we mention earlier large graphs are unfeasible to store them in a computer and to calculate their properties. We consider the properties of the large graph as a *ground truth*. If we want to analyze a large graph we need to sample a small portion of it. But how can we obtain a sample and what information can the sample give to us?

Sampling algorithms can be the solution of our problem. Sampling algorithms can sample a large graph with different way .We think the sampling process as a *crawling*. It is a way that our algorithm starts from a *seed/node* and explores the graph.

The process of crawling plays initial role. There are two techniques which can use in order to crawl a graph; the *graph traversal technique* and the *random walk technique*. With *graph traversal technique* every node can be visited once. It starts from a node and explore the entire graph until it has sample all the nodes. The sample will have all the explored nodes only once.

In contrast with *random walk* revisiting nodes is possible. With random walk our priority is to sample a portion of nodes. The nodes we have revisit will be included in the sample.

Graph traverse algorithms are biased to high degree nodes such as the Breadth-First-Search, Greedy, Lottery cannot give us accuracy for the properties of the large graph [4].

On the other hand random walk algorithms such as the Metropolis hasting Random Walk and Random Walk Re-Weighted can give us an unbiased sample and we can make better predictions [1].

In sampling one of the major problems is the *bias* which it means that the sample cannot help us to make accurate prediction for the properties of the large graph. So we cannot make properly estimations of the graph properties. The challenge is to crawl those nodes who are going to help us to get an accurate insight of the large graph. Note that, the sampled network is smaller, so there is a scaling effect on some of the statistics. For instance the average degree of our sampled is smaller. To overcome bias problem it is really difficult.

2.1 Graph Properties

Let $G_d = (V, E_d)$ be a directed social graph, V is the set of nodes which is the number of users and E_d the edges that links users. The edges of a node $u \in V$ can be indegree and outdegree. An edge $(u, w) \epsilon E$ is the link between the nodes. We can denote the indegree of a node $u \in V$ as $d_{in}(v)$ and the outdegree nodes as $d_{out}(v)$. In order to calculate the mean degree we need to find the number of outdegree and indegree and to divide with the number of the nodes.

$$a = \frac{2\,|E_d|}{u} \tag{1}$$

Another approach which is very similar with the degree. Is the node degree distribution. Is a metric which is very important and give us very good insight for the graph. By calculating the degree distribution of a graph we can understand which user interact with other users.

3. RELATED WORK

A lot of studies assessed the quality of sampling methods based on sampling algorithms. Faloutsos and Leskovec [2] attempt to estimate the properties of the large graph by analyzing the properties of the sample and their sampling goals was *back in time* and *scale down*. Scale down is a sample with similar properties or scaled down properties of our initial graph. Back in time is a sample which represents the initial graph back in time when it had a smaller size. Their result was that the random walk(RW) performed best for scale down and Forest fire for back in time goal.

A different approach in [4] where sampling methods have been used in order to simulate the process of a web crawler. They used graph traverse techniques to explore large OSNs. They achieve to explore a big portion of the graph and to obtain samples. But they couldn't make accurate estimations.

Gjoka [1] research focused on how to obtain unbiased sample from Facebook their results was that Metropolis-Hasting-Random-Walk(MHRW) and Random walk re-weight (RWRW) performed remarkably well and close to the ground truth for certain properties.

A similar approach [3] on sampling social graphs showed the MHRW and Frontier sampler(FS) can estimate the node degree distribution and the clustering coefficients accurately although the results depend on the structure of the network.

Also, those researches have accomplished to give us some good results about sampling methods and a lot of new sampling algorithms have implemented according to our target which can give a representative sample. On the other hand, new problems appeared such as the kind of the dataset that researchers used seems to affect the results [3]. Since the research continues new algorithms keep developing but they have not tested in deep.

In this paper, our goal is to simulate how to crawl a large graph. We will simulate a crawl with BFS and how we can obtain a sample of an OSN how good is the sample. We will follow the same procedure as in [4]. We will compare BFS with another algorithm Metropolis Hasting Random walk. We will crawl the same amount of nodes and we will evaluate the sample which properties of the large graph retains.

Table 1: Notation

Notation	Definition
Q_{crawl}	**the set of nodes we will process**
V_{seen}	**the set of nodes we have seen**
V_{crawl}	**the set of nodes we have crawled**
Out_u	**the neighbors of node u**
u	**the initial node we choose**
prt	**probability generated uniform**
w	**node of the neighbors of u**
G_u	**nodes of the whole graph**

4. SUGGESTED APPROACHES

In previous research they used the number of seeds and the portion of the sample as a factor in order to estimate the average degree of a social network [4].

A different technique we can observe in the research [1] where they used the number of iterations as a factor in order to understand how the algorithms converges in order to have accurate estimations about the degree distribution.

In our approach we will

4.1 Sampling Algorithms

Conceptually crawling techniques can be classified as graph traversal techniques and random walk:

4.2 Graph traversal techniques

Depth first search is very popular graph traverse technique which select arbitrary nodes and explore them to the depth before backtracking.
Forest fire (FF) picks a node random and uniformly and start 'burning' out going links and their nodes. This algorithm performed best for back in time sampling goal [2].
Snowball(SBS) a sampling method which is very similar to BFS. This algorithm selects a number of seeds and adding their neighbors in the sample to the max depth. For every iteration samples nodes that are not already in the sample to avoid repeating. It's very effective and only with small portion 1% can be accurate for the degree distribution [4].

4.3 Random walk techniques

As more stochastic and complex approach considered an exploration techniques like random walk(RW) which pick a node randomly and uniformly and start walking on the graph. With the walk we mean sampling nodes [2]. In each iteration it selects a neighbor of the current node to visit. The random walk is biased to high degree nodes.
Random walk with restart (RWR)is a node selection algorithm based on the random walk. The key idea of this algorithm is that it starts sampling nodes and restart walking from a fixed node based on a give probability.
Random walk with random jump (RWJ) the main idea of this algorithm is the desire of simulation random walk on a directed graph. As in directed networks the random walk can stuck with this implementation when it stuck it jumps to an arbitrary node with uniform probability.

4.4 Assumptions

Before we continue we make some assumptions that we have prior knowledge in order to evaluate the efficiency of our sampling algorithms. We have knowledge of the links between the users in order to calculate the ground truth. The sample that we are going to gather with every algorithm will be same in order to evaluate their performance. We will use one seed for breadth-first-search as and one seed for Metropolis hasting random walk.

4.5 Factors

While we try to crawl an OSN there are several factors we need to consider before we proceed:

- Number of seeds
- Protected users
- Sample size

Seeds: The number of seeds seems to play a crucial role in the process of crawling. Seeds are the number of nodes which we select randomly in order to start the process of crawling. The main idea of sampling, seeds can be chosen randomly or voluntarily. In this paper, we will one for the BFS which are chosen randomly and for Metropolis-Hasting-Random Walk. Although previous research shows that the number of seeds does not affect our sample. [4]. In our research,we consider the seeds as our prior knowledge. We simulate a web crawler so we can pick from the beginning, which users will be our seeds on process of crawling.
Protected users: OSNs the last years have given users the option of privacy. This kind of users can be a problem for the crawler because they cannot crawl private users in other research [4] considers those users as a 'blackhole'. Typically means that is a node without outgoing links. Sampling algorithms like the BFS don't seem to hurt by black holes. But other sampling algorithms like the Metropolis Hasting Random walk, maybe get stuck. That's the reason that we consider every node with zero degree as bidirectional. So Metropolis Hasting Random walk cannot stuck and always return at it's previous position.
Sample size: We need the sample that can give us insight and retain the properties of the largest graph. We are given a large static directed graph G of u nodes. The goal is to crawl a sample graph S on u' nodes, that will be most similar to G. The sample size that we are going to crawl with both algorithms is the 10 percent of the whole graph.

4.5.1 Breadth First Search

Breadth first search (BFS) is widely popular and has been used for network analysis. This algorithm, it starts with random selecting a seed/node. In every iteration it selects to sample the earliest explored but not yet visited node. There are two queues the one queue is Q_{crawl} which are the nodes of the sample that we have processed and in the other queue V_{seen} that have explored . Initially, it adds the seeds in all queues and in every iteration the first node u in queue Q_{crawl} it pops to the queue V_{seen} which is our sample. All the neighbors of node u move to the queue V_{seen} and Q_{crawl} unless they are already in. The process is continued until we have sampled the whole graph. It's known that BFS is biased to high degree nodes and has been compared with other sampling methods in [4, 3] its performance was quite poor compared with the other sampling methods.

Algorithm 1 Breadth First Search

```
$Q_{crawl} \leftarrow V_{seed}$
$V_{seen} \leftarrow V_{seed}$
while $Q_{crawl} \neq \emptyset$ do:
select and remove u for $Q_{Crawl}$
    $Out \leftarrow crawl(u)$
    for all $u' \in Out_u$ and $u' \notin V_{seen}$ do:
        insert $u'$ into $Q_{crawl}$ and $V_{seen}$
    end for
end while
```

4.5.2 Metropolis - Hasting Random Walk(MHRW)

Metropolis - Hasting Random Walk (MHRW) MHRW is an exploration technique, it based on Random Walk(RW)

sampling technique. MHRW is least bias because it is Markov-Chain Monte Carlo (MCMC) algorithm. It random sampling nodes according to the degree probability of the nodes. In MHRW a proposal function is designed based on the probability distribution. By randomly accepting or refusing the proposal, the proposal function changes the transition probabilities, making the sample converge to the probability distribution [3].
The Metropolis - Hasting algorithm starts by sampling a random seed with non zero neighbors. We define the proposal function as k_u which is the degree of the node u. Next it chooses random one w of the neighbors of the node u. The MHRW generates a probability from uniform distribution (0,1). If probability $< k_u/k_w$ the proposal is accepted and added in the sample. Else remain at the node u.
The whole process of MHRW is that accept a node of smaller degree, and usually reject sampling nodes with high degree and with this way it eliminates the bias towards high degree nodes.

Algorithm 2 Metropolis - Hasting Random Walk

```
$u$ ←initial node
while Sample size not met do:
    select $w$ uniform at random from $Out_u$
    pr = generate probability uniform(0,1)
    if $pr \leq k_u/k_w$ then :
        $u \leftarrow w$
    else
        Stay at u
    end if
end while
```

5. DATASETS

Epinions: This dataset based on the relationship of users according to their trust. It's a website which users review products and select which user to trust. It's an idea that who-trust-who. We download this dataset from the Stanford. It has been used for previous researches [2].
Slashdot: Slashdot is a technology-related news website known for its specific user community. The site contains a feature Slashdot zoo, which allows users to tag each other as friends. The dataset contains links of users,which are friends obtained in 2008.

Table 2: Datasets

Dataset	E-pinions	Slashdot
Nodes	75879	77360
Edges	508837	905468
Average CC	0.1378	0.0555
Average Degree	13.411	23.404
Nodes in largest SCC	32223	70355

5.1 Evaluation

For the evaluation of the algorithms, we will consider how many nodes and edges our algorithms can discover by crawling just a small portion of our graph. It's actually a metric which can evaluate how much closer is the size of our sample compared with the whole graph. We divide this metric in *node coverage* and *link coverage* [4]:

- Node coverage (NC): The nodes that our crawler has sampled divided by the number of nodes of the whole graph $\frac{|V_{seen}|}{|V|}$.
- Link coverage (LC): Very similar to the *node coverage* is the number of edges of our sample divided by the number of links of the whole graph $\frac{|E_{seen}|}{|E|}$.
- Average Degree : We will

5.2 Crawling Efficiency

As we can observe in table 3 and table 4 both crawlers performed well exploring the same amount of nodes and links. Both crawlers were able to discover a big portion of the graph by just crawling only the five percent of the graph. There is a balance between the nodes and the links that our algorithms discovered. Although as we continue to crawl the distance between the node coverage and link coverage changes. We started crawling from five percent until thirty percent of our graphs.

Dataset	NC	LC
Epinions	35%	43%
Slashdot	63%	35%

Table 3: RW:Node link coverage with 5%

Dataset	NC	LC
Epinions	34%	43%
Slashdot	60%	34%

Table 4: BFS:Node link coverage with 5%

5.2.1 *Node Coverage*

We cannot observe any big difference between the two samples. The major difference that we can observe in figure 1 random walk was able to explore more nodes than Breadth-first-search. This must be the result of the jumps that random walk performed when it was stuck. The nodes that it crawls after every jump were nodes with high degree. random walk performed better than BFS. Node/link coverage is sensitive to the number of jumps.

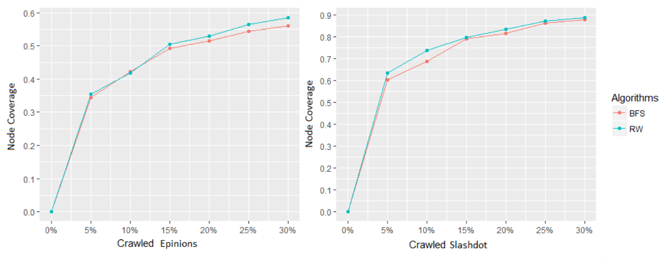

Figure 1: Node coverage Datasets Random Walk and BFS

In figure 2 shows the results of crawling with random walk on the two datasets. The big difference of the node coverage in Slashdot compared with the Epinions is a result of its high mean degree of the graph. The mean degree of Slashdot is (23.4) and of Epinions is (13.3). That's explains the high node coverage in Slashdot compared with the Epinions. We

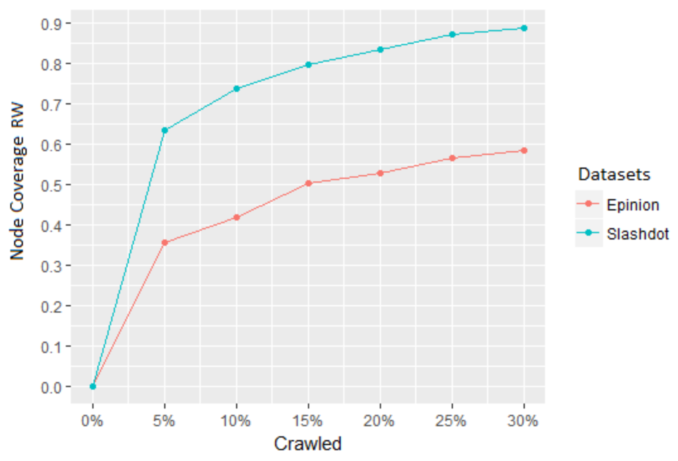

Figure 2: Node coverage Random Walk

obtain the same difference by crawling with BFS.

5.2.2 *Link Coverage*

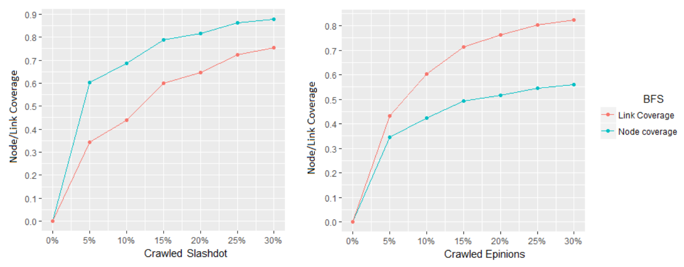

Figure 3: Node/Link coverage BFS

In Figure 3 is the number of nodes and links that BFS explored by crawling. Evidently the number of links that we explored is higher for the Epinions than the Slashdot. The Epinions graph has a much smaller, tightly coupled core compared to the Slashdot graph. Also the average clustering coefficients of the Epinions graph is (0.139) compared to the Slashdot graph (0.05) the difference is not big, but that means that the nodes of Epinions graph share more edges.

5.3 Sample size

In order to have accurate estimation and to evaluate the algorithms we crawled the same number of nodes. The portion that both algorithms explored is relatively the same. Random walk needed extremely more time than the BFS. We also kept track of the jumps that the random walk performed. We consider every jump as a 'blackhole' or as a node which have already crawled. The jump can give us some further details about the structure of the graph.
In figure 4 we can see the big difference of the jumps that random walk performed for every dataset. As we crawled more of the graph the number of jumps increased. For the Epinions dataset random walk performed almost two thousand jump in order to collect an efficient sample. random walk could easily discover communities. But for the Slashdot dataset the number is extremely large. The Slashdot is the dataset with the most strong connected nodes. The high number of jumps shows that our crawler attempted to crawl nodes which has already collected or that the portion of nodes without link is substantial.

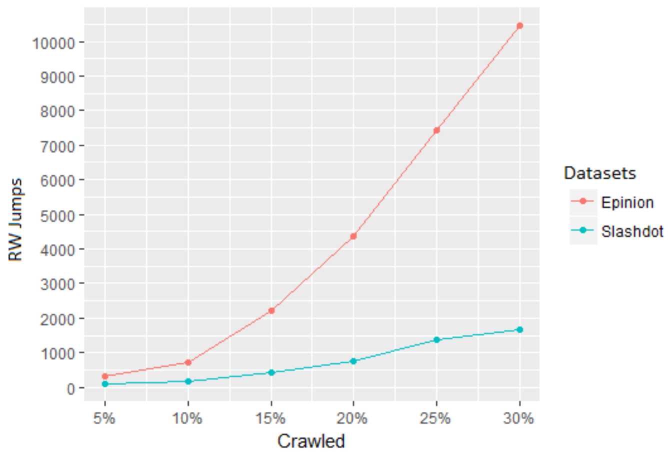

Figure 4: Jumps performed by Random walk for both graphs

5.4 Crawling Bias

In this section we will evaluate the properties of the sample compared with the ground truth. As a ground truth, we consider the properties of the large graph the average degree and the average clustering coefficients.

- Absolute degree (AD): The average degree of the sampled abstracted by the average degree of the nodes of the whole graph $|a_{crawled} - a|$.
- Absolute clustering coefficients (ACC): The average clustering coefficients of our sample abstracted by the average coefficients of the whole graph $|\overline{C}_{crawled} - \overline{C}|$.

5.4.1 *Absolute degree*

In figure 5 we evaluate how bias was our sample. Both algorithms are biased to high degree nodes. They both explored big portion of the graph in order to approach the ground truth. BFS was the algorithm which performed worst in both datasets. Random walk and BFS started by collecting high degree nodes, but both seem to converge as we continue to crawl. In figure 6 random walk starts crawling nodes with higher degree than the BFS but as it crawls more it gets better estimation than the BFS.
By crawling just the thirty percent of the graph, none of our algorithms couldn't estimate accurately the average degree of the largest graph. Breadth first search is more biased to high degree nodes than the random walk.

5.4.2 *Absolute clustering coefficients*

Next we evaluate the average clustering coefficients of our sample. As we mention earlier as a metric we consider the Absolute clustering coefficients. In figure 7 and figure 8 we can observe that both algorithms cannot estimate the clustering coefficients accurate. This happens because both algorithms are bias to high degree nodes. So the possible intersections between the nodes of our sample are higher than

the ground truth. For both datasets, as we crawl larger portion of the graph the distance between our prediction and the ground truth seems to increase. Our algorithms failed to estimate the average clustering coefficients.
The only observation, it can be done according to our results is that the best prediction was when we crawled the ten percent of the large graphs. BFS and RW are really close to the ground truth compared our next predictions. RW performed better than BFS and it was closer to the ground truth. The performance of the random walk is due to its behaviour to explore the graph in depth in contrast with the BFS which oversampled high degree node.

6. CONCLUSION

We tested two different algorithms in two different datasets. We evaluate their performance by crawling the same portion of the graph. We compared the properties of our sample with the properties of the large graph. We investigate how many times a random walk algorithm can 'stuck' during the process of the crawling.
Our implications will probably help future studies. The process of crawling is very interesting procedure. Major problem is still the bias of high degree nodes . A random walk technique performed better than the BFS. But we still couldn't eliminate the bias problem. We made the key observation that with small sample size we have the better estimation of average clustering coefficient. After we crawl even the thirty percent of our social graph we cannot estimate accurately the average degree of the large graph. Also

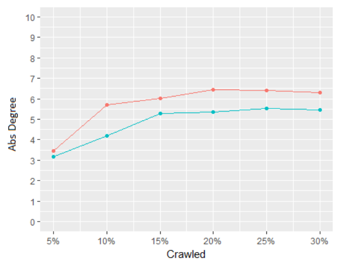

Figure 5: Absolute degree Slashdot.

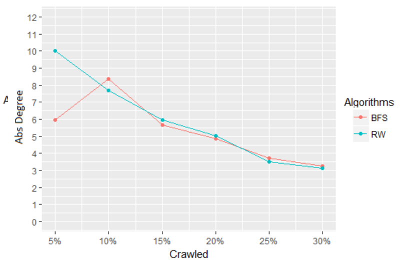

Figure 6: Absolute degree Epinions.

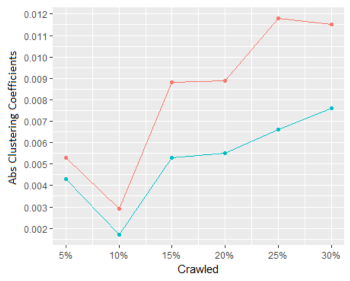

Figure 7: Absolute clustering coefficient Slashdot.

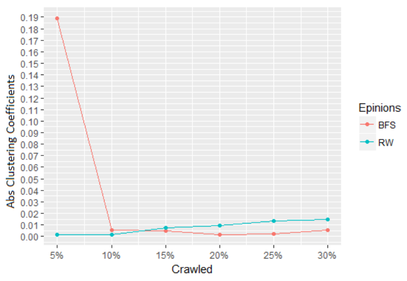

Figure 8: Absolute clustering coefficient Epinions.

we figure out that the bias depends on the structure of the network. The number of jumps that random walk performed can help future studies.
A good question is that if the number of jumps can used as the number of seeds for graph traversal techniques. And how our estimation would change the results. For instance, if the BFS crawled the same portion of the graph based on the number of jumps as seeds. In addition the time complexity is something which did not mention. Moreover, further investigation in larger graphs could give us more information about crawling and multiply tests can be done.

7. REFERENCES

[1] M. Gjoka, M. Kurant, C. T. Butts, and A. Markopoulou. Walking in facebook: A case study of unbiased sampling of osns. In *Infocom, 2010 Proceedings IEEE*, pages 1–9. IEEE, 2010.

[2] J. Leskovec and C. Faloutsos. Sampling from large graphs. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 631–636. ACM, 2006.

[3] Z. Z. T. X. L. J. P. H. B. D. X. L. Tianyi Wang, Yang Chen. Understanding graph sampling algorithms for social network analysis. In *Distributed Computing Systems Workshops (ICDCSW), 2011 31st International Conference*, pages 1–6. IEEE, 2011.

[4] S. Ye, J. Lang, and F. Wu. Crawling online social graphs. In *Web Conference (APWEB), 2010 12th International Asia-Pacific*, pages 236–242. IEEE, 2010.