A two-phase Sampling based Algorithm for Social Networks

Zeinab S. Jalali, Alireza Rezvanian, Mohammad Reza Meybodi
Soft Computing Laboratory, Computer Engineering and Information Technology Department
Amirkabir University of Technology (Tehran Polytechnic)
Tehran, Iran

***Abstract*— in recent years, the data used for analysis of social networks become very huge and restrictive so that it can be used an appropriate and small sampled network of original network for analysis goals. Sampling social network is referred to collect a small subgraph of original network with high property similarities between them. Due to important impact of sampling on the social network analyses, many algorithms have been proposed in the field of network sampling. In this paper, we propose a two-phase algorithm for sampling online social networks. At first phase, our algorithm iteratively constructs several set of minimum spanning tree (MST) of network. In the second phase, the proposed algorithm sorts vertices of MSTs and merge them to form a sampled network. Several simulation experiments are conducted to examine the performance of the proposed algorithm on different networks. The obtained results are compared with counterpart algorithms in terms of KS-test and ND-test. From the results, it can be observed that the proposed algorithm outperforms the existing algorithms.**

Keywords—online social networks; social network analysis; network sampling; spanning tree

I. Introduction

Today online social networks (OSN) are become the one of the most important parts of human lives. Many techniques are developed for social network analysis by measuring the characteristics of OSN to study the properties of these networks. Because of large scale and access limitation of these networks, it is easy to study these networks completely [1], so sampling a representative graph from original online social networks is essential to study the characteristics of the networks [2-5].

A social networks can be represented as a graph $G= \langle V, E\rangle$ with a set of nodes V and edge-set E where nodes represent users and edges represent a kind of relationship between users in a social network. A representative sample of large social network is a small scale of graph that preserves features of initial graph. Using this sample, one can study the important features of initial graph such as distribution, user activities and user connectivity [6-7].

Sampling techniques are used across any settings. Data mining, information retrieval, simulation, experimentation, measuring performance of protocols, viral marketing and fraud detection are some examples of these settings [8]. Several forms of sampling techniques have been studied in the literature. There exists some classification for sampling techniques.

- First view classifies sampling algorithms into node sampling; edge sampling and topology based sampling [8]. In node sampling, nodes are sampled independently at random; breadth first search (BFS) and Metropolis hasting random walk (MHRW) are examples of node sampling based algorithm. In edge sampling, edges of graph are sampled independently at random; frontier sampling is an example of this algorithm. In topology based sampling, selection of a node or an edge depends on topology of initial graph; snow ball sampling [9], Forest Fire and Random walk based sampling algorithms [10-12] are considered as topology based sampling.
- Second view classifies sampling algorithms into graph traversals and random nodes [13]. Random walk based sampling algorithms are examples of this group. In second category, each node is visited at most one time and there is no replacement during sampling. Breadth first sampling, Forest Fire [14], Snowball sampling and respondent driven sampling [15] are examples of these group.
- Third view classifies sampling algorithms into scale down and back in time goals [16]. In former algorithms, goal is to find a sample graph with the most similarity to initial graph; metropolis random walk is an example of this algorithm. In latter algorithms, goal is to travel back in time and find past version of initial graph when it was the same size as sample graph. Forest fire is an example of this algorithm.

There exist several algorithms that use mentioned algorithms to find a good sample from online social networks. These algorithms can be used by connected or clustered, directed or undirected and weighted or unweight graphs.

In the rest of this paper, in section 2 we describe the proposed sampling algorithm. In section 3, we express the simulation results, and section 4 concludes the paper.

II. Proposed Sampling Algorithm

The proposed sampling algorithm consists of two phases for sampling online social networks. At first phase, our algorithm iteratively constructs several set of minimum spanning tree (MST) of given network. In the first phase, our proposed algorithm randomly constructs a set of spanning trees at each iteration by starting at a randomly selected node v_i (as root node). Let $G_s=\langle V_s, E_s\rangle$ be a spanning tree from connected graph $G=\langle V, E\rangle$, where $V=\{v_1, v_2 ,..., v_n\}$ is the set of nodes and $E=\{e(v_i ,v_j)\}\subseteq$ V×V denotes the set of edges with the function $f:G\rightarrow G_s$. Using this function, G_s should be connected, G and G_s should have the same node set and $|E_s|=|V_s|-1$ where $|E_s|$ and $|V_s|$ is the cardinality of edge-set and vertex-set of graph respectively [17].

After constructing set of spanning trees, in the second phase of the proposed algorithm a rank is assigned to each edge appearing in the spanning trees based on the number of times that spanning trees along which that edge has been appeared. The sampled network now can be constructed by considering a sub-graph of the input graph whose node set contains with *m* percent of the highly ranked edges. The main idea behind using spanning tree as promising parts of graph is to obtain central nodes and edges of graph among all nodes and edges of graph because one can be spread to main part of graph through these nodes/edges. Moreover, by applying several spanning trees between any desired nodes we can reach to more diverse and proper samples of initial graph. **Error! Reference source not found.** gives the pseudo code of the proposed sampling algorithm.

```
Algorithm 1: Sampling graph with spanning tree
Input: Graph $G=\langle V, E\rangle$,
  k: Number of computed spanning trees,
  m: Number of nodes in the sampled network,
Output: Sampled graph $G_s=\langle V_s, E_s\rangle$
Begin
  Let $\tau_t$ denotes the spanning tree at iteration $t$;
  While ($t < k$)
      Select $v_s$ randomly as initial node of spanning tree;
      Find the spanning tree $\tau_t$ from $v_s$;
      $t \leftarrow t+1$
  End While
    Assign a rank to each edge that has been occurred in the spanning
  trees; The rank of an each is found according to the number of time that
  the edge has occurred in the spanning trees.
    Generate sampled network $G_s$ by considering a subgraph of the input
  graph which contains $m$ percent of the highly ranked edges
End Algorithm
```

Figure 1. The pseudo code of the proposed Sampling graph with a set of spanning tree

III. Simulation Results

To show the performance of the proposed algorithm several simulation experiments were conducted on several graph instances. The graph instances that are used in this set of experiments are shown in Table 1.

Table 1. Description of test networks

Network	Node	Edge	Description
Epinions	75879	508837	A friendly place to get answers to even the most basic questions
Cit-HepPh	34546	421578	Physics theory citation network
Slashdot0902	82168	948464	A snapshot of Slashdot Zoo social network from February 0 2009.

A. Evaluation metrics

In this paper, we use *Kolmogrov-Smirnov Test* (K-S test), and *Normalized-Distance*) for performance studies. They are described in the rest of this subsection.

- Kolmogrov-Smirnov D-Statistic is one of the statistical test algorithms used for assessment the distance between two cumulative distribution functions (CDF). *D* is a measure for acceptability between original distribution and estimated distribution. The result of this test can be relatively employed for comparison. In other words, the result of this test is a value between 0 and 1. As closer as it is to zero, both distributions will have a greater similarity; and as closer as it is to unit, the two distributions will show a greater discrepancy [43]. This measure has been defined as:

$$D = \max\left|F'(x)-F(x)\right| \tag{1}$$

- Normalized L_1 distance: In some cases, for evaluation we will need to measure the distance between two positive m-dimensional real vectors F(x) and F'(x) such that F(x) is the true vector and F'(x) is the estimated vector. For example, to compute the distance between two vectors of eigenvalues. In this case, we use the normalized L_1 distance

$$L_1(F'(x),F(x)) = \frac{1}{m}\sum_{i=1}^{m}\frac{\left|F(x_i)-F'(x_i)\right|}{F(x_i)} \tag{2}$$

B. Experimental results

To show the performance of the proposed algorithm, several experiments are conducted on the networks described in Table 1. All experiments for the proposed sampling algorithms are evaluated in terms of KS-test and ND-test, and then results are compared with classic sampling algorithms such as Random Node Sampling (RNS) [18], Random Edge Sampling (RES) [8], Random Walk (RW) [19] and Metropolis-Hastings Random Walk (MHRW) [20] for different sampling rates.

C. Experiment I

This experiment is executed to study the minimum number of required spanning trees. The proposed algorithm have launched on different test networks to calculate number of required spanning trees. For three mentioned datasets in table 1, 0 to 100 spanning trees has been produced. Results show that for sampling rate of 30%, about 50 spanning trees is needed to meet desired sampling criteria. Figure 2-5 show the results.

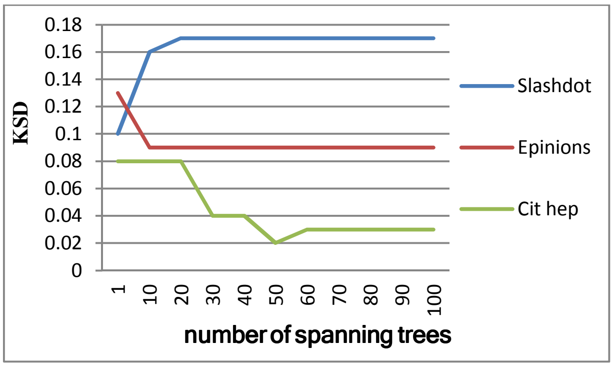

Figure 2 : KSD for CCD with SST with sampling rate of 30%

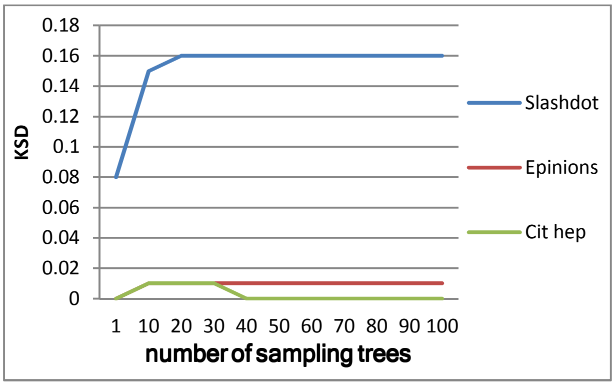

Figure 3 : SDD for DD with SST with sampling rate of 30%

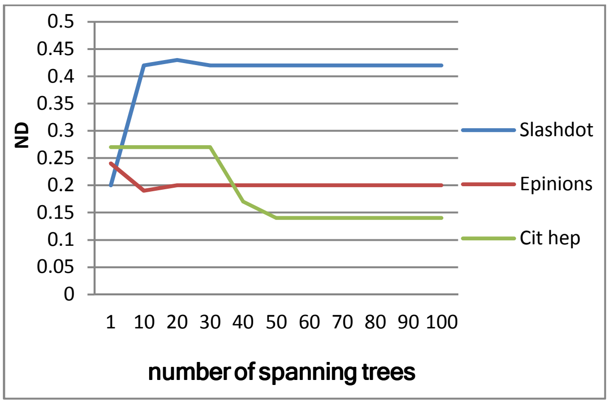

Figure 4 : ND for DD with SST with sampling rate of 30%

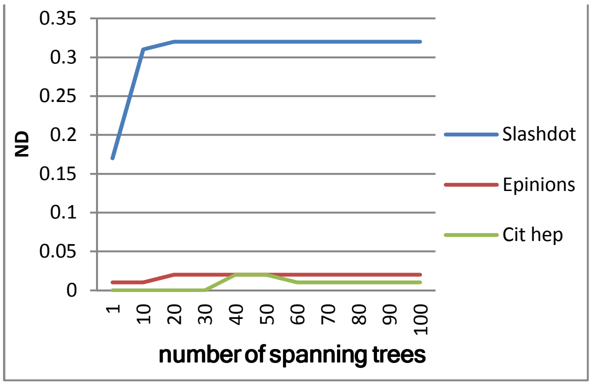

Figure 5 : ND for CCD with SST with sampling rate of 30%

D. *Experiment II*

This experiment is carried out to compare the performance of the proposed algorithm with different sampling rates. The sampling rates are changed from 0.05 to 0.30. Unlike many other studies whose sampling rate are in a large value, the result of our proposed algorithm for lower sampling rates like 5-10% are acceptable, which means the cost of our sampling algorithm is lower than other algorithms. Table 6 – 9 shows the results.

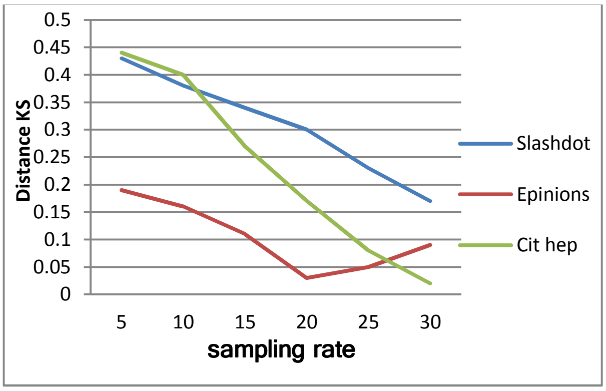

Figure 6: KSD for DD with SST with sampling rate of (5-30)%

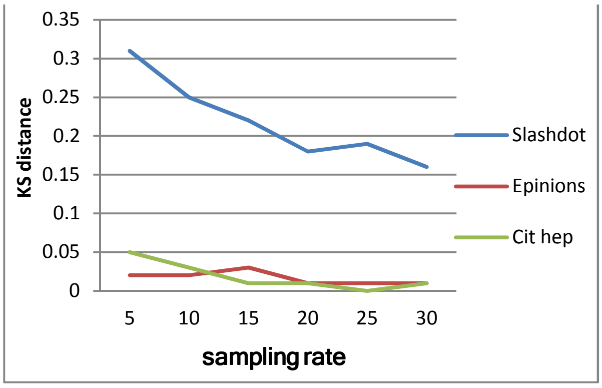

Figure 7 : KSD for CCD with SST with sampling rate of (5-30)%

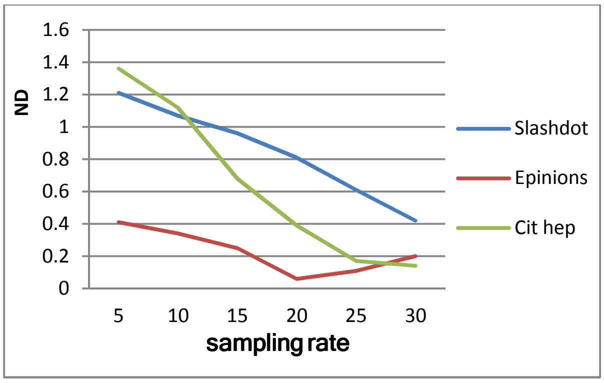

Figure 8: ND for DD with SST with sampling rate of (5-30)%

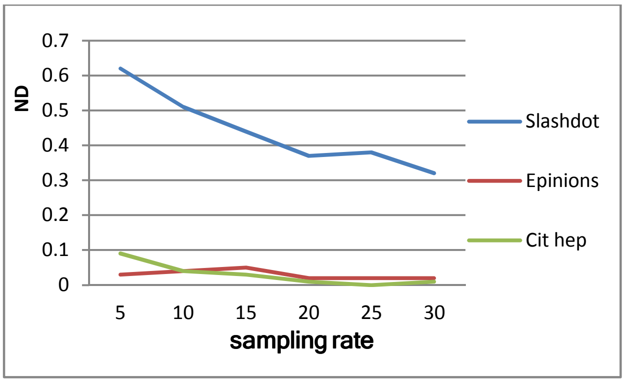

Figure 9: ND for CCD with SST with sampling rate of (5-30)%

E. *Experiment III*

This experiment is designed for comparing performance of the proposed sampling algorithm with spanning tree (SST) with other classic sampling algorithms like Random Node Sampling (RNS), Random Edge Sampling (RES), Metropolis-Hastings Random Walk (MHRW) and random walk (RE) for the sampling rate of 15. The results for Random node, Random edge, Random Walk and metropolis hasting random walk are taken over 30 runs. Then for comparing results we use t-test to show the significance of results. In this experiment the results of proposed algorithm are compared with results of other algorithms in term of both KS and ND for both degree distribution (dd) and clustering coefficient (cc). Superiority of SST is shown by checkmark (√) and equality is shown by equal mark (~), otherwise it works worse (×). The experimental result shows that the proposed algorithm works better than other sampling algorithms in terms of KS-test and ND-test. Results of this experiment are shown in table 2 – table4.

Table 2: comparison of proposed algorithms for Cit-hep with sampling rate of 15%

		RN	RW	RE	MHRW	SST
KS	dd	0.81±0.01	0.17±0.00	0.07±0.00	0.07±0.02	0.27±0.00
		√	×	×	×	
	cc	0.05±0.03	0.02±0.00	0.00±0.00	0.05±0.01	0.01±0.00
		√	√	×	√	
NV2	dd	0.00±0.17	0.32±0.01	0.24±0.01	0.19±0.02	0.68±0.00
		√	×	×	×	
	cc	0.09±0.00	0.04±0.00	0.01±0.00	0.08±0.00	0.03±0.00
		×	√	×	√	

Table 3 : comparison of proposed algorithms for Epinion with sampling rate of 15%

		RN	RW	RE	MHRW	SST
KS	dd	0.18±0.02	0.17±0.00	0.45±0.00	0.39±0.02	0.39±0.00
		×	√	√	√	
	cc	0.10±0.03	0.02±0.00	0.10±0.00	0.08±0.01	0.08±0.00
		√	×	√	√	
NV2	dd	0.34±0.06	0.32±0.01	0.71±0.01	0.58±0.02	0.58±0.00
		×	√	√	√	
	cc	0.18±0.00	0.04±0.00	0.17±0.00	0.14±0.00	0.14±0.00
		√	×	√	√	

Table4: comparison of proposed algorithms for Slashdot with sampling rate of 15%

		RN	RW	RE	MHRW	SST
KS	dd	0.0.79±0.01	0.04±0.00	0.43±0.00	0.39±0.02	0.34±0.00
		√	×	√	√	
	cc	0.28±0.02	0.05±0.00	0.35±0.00	0.30±0.01	0.22±0.00
		√	×	√	√	
NV2	dd	2.16±0.03	0.10±0.01	0.76±0.01	0.66±0.04	0.96±0.00
		√	×	√	√	
	cc	0.56±0.00	0.10±0.00	0.71±0.00	0.61±0.00	0.44±0.00
		√	×	×	×	

IV. CONCLUSION AND FUTURE WORK

In this paper, we proposed an algorithm for sampling social networks based on spanning trees. Our algorithm tried to collect set of nods and edges for constructing a sampled network based on the fact that edges appearing in spanning trees can represent the most structure information of a network. By this idea, we present a sampling algorithm that is efficient in sampling online social networks. The performance of the proposed sampling algorithm was investigated by executing several experiments on different data sets. The results compared with classic sampling algorithms such as node sampling, edge sampling, random walk and metropolis hasting random walk. The experimental results showed that the proposed sampling algorithm performs better than other sampling algorithms in terms of KS-test and ND-test.

The new proposed algorithm can be extended in many directions, which could be the future work. Firstly, spanning tree can be applied on parts of graph instead of whole graph to save time and memory. Secondly, the algorithm can be contributed to be applied on weighted graphs.

REFERENCES

[1] M. Huisman, "Imputation of missing network data: Some simple procedures," Journal of Social Structure, vol. 35, no. 4, pp. 1-29, 2009.

[2] M. Papagelis, G. Das, and N. Koudas, "Sampling Online Social Networks," IEEE Transactions on Knowledge and Data Engineering, vol. 25, no. 3, pp. 662-675, 2013.

[3] M. Gjoka, C. T. Butts, M. Kurant, and A. Markopoulou, "Multigraph sampling of online social networks," IEEE Journal on Selected Areas in Communications, vol. 29, no. 4, pp. 1893-1905, 2011.

[4] E. Volz, and D.D. Heckathorn, "Probability based estimation theory for respondent driven sampling," Journal of Official Statistics-Stockholm, vol. 24, no. 1, p. 79, 2008.

[5] A. Rezvanian, M. Rahmati, M. R. Meybodi, “Sampling from complex networks using distributed learning automata,” Physica A: Statistical Mechanics and its Applications, vol. 396, 224-234, 2014.

[6] U. Pfeil, R. Arjan, and P. Zaphiris, "Age differences in online social networking–A study of user profiles and the social capital divide among teenagers and older users in MySpace," Computers in Human Behavior, vol. 25, no. 3, p. 643–654, 2009.

[7] C. Wilson, B. Boe, A. Sala, K. P. N. Puttaswamy, and B. Y. Zhao, "User interactions in social networks and their implications," in fourth ACM European Conference on Computer Systems, 2008.

[8] N. K. Ahmed, F. Berchmans, J. Neville, and R. Kompella, "Time-based sampling of social network activity graphs," in Eighth Workshop on Mining and Learning with Graphs, 2010.

[9] L. Goodman, "Snowball sampling," The Annals of Mathematical Statistics, vol. 32, pp. 148-170, 1961.

[10] L. Lov´ asz, “Random walks on graphs: A survey,” Combinatorics, Paul Erdos is Eighty, vol. 2, no. 1, pp. 1–46, 1993.

[11] M. R. Henzinger, A. Heydon, M. Mitzenmacher, and M. Najork, “On near-uniform URL sampling,” in Proc. 9th Int. Conf. on World Wide Web, Amsterdam, Netherlands, 2000.

[12] M. Kurant, M. Gjoka, C. T. Butts, and A. Markopoulou, “Walking on a Graph with a Magnifying Glass: Stratified Sampling via Weighted Random Walks,” in Proc. ACM SIGMETRICS, 2011.

[13] towards unbiased sampling

[14] M. Kurant, A. Markopoulou, and P. Thiran, “Towards unbiased BFS sampling,” IEEE Journal on Selected Areas in Communication, vol. 29, no. 9, pp. 1799-1809, 2011.

[15] J. Leskovec, and C. faloutsos, "Sampling from Large Graphs," in twelfth ACM SIGKDD International Conference of Knowledge Discovery and Data Mining, 2006.

[16] A. Rasti, M. Torkjazi, R. Rejaie, N. Duffield, W. Willinger, and D. Stutzbach, "Respondent-driven sampling for characterizing unstructured overlays," in IEEE INFOCOM, 2009.

[17] J. A. Torkestani, "Degree constrained minimum spanning tree problem: a learning automata approach," The Journal of Supercomputing, vol. 64, no. 1, pp. 226-249, 2013.

[18] A. S. Maiya, and T. Y. Berger-Wolf, "Benefits of Bias: Towards Better Characterization of Network Sampling," in Seventeenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2011

[19] L. Lov´ asz, "Random walks on graphs: A survey," Combinatorics, Paul Erdos is Eighty, vol. 2, no. 1, p. 1–46, 1993.

[20] F. Murai, B. Riberio, D. Towsley, and P. Wang, "On Set Size Distribution Estimation and the Characterization of Large Networks via Sampling," IEEE Journal on Selected Areas in Communications, vol. 31, no. 6, pp. 1017-1025, 2013.