An Efficient Algorithm for Sampling of a Single Large Graph

Melkaukan analisis pada subgraph lebih efisien daripada pada graph besar. Tapi bagaimana cara menentukan subgraph yang bagus

Vandana Bhatia[1] and Rinkle Rani[2]

Department of Computer Science and Engineering
Thapar University
Patiala, India
vandana.bhatia@thapar.com[1], raggarwal@thapar.com[2]

***Abstract*— Graph Databases offer a very influential way to provide an instinctual representation for many applications spanning from social networks, web networks to biological networks. In the current era of big data, the size of the graph is increasing exponentially. It is difficult for the conventional machines to analyze a whole graph. To overcome this, the characteristics of the large graphs are estimated via sampling in order to identify trends and patterns in the large graph. The existing sampling techniques such as random node and random walk do not provide consistent efficiency over the graphs. In this paper, an efficient sampling algorithm named Influence sampling (IS) is proposed which sample the graphs by analyzing the degree of the vertices of the graph such that the most influential vertices remain in the graph sample. The experiments are performed over three real life datasets and the performance is compared with the three existing sampling algorithms. It is shown that IS performs well in the terms of accuracy.**

Keywords*— *Sampling; large graphs; Graph Mining; Random Walks, Node Degree

I. INTRODUCTION

In the last decade, the data has grown at a tremendous rate. Representing the data as graph to understand the relationship between data objects is gaining high interest. But with the increase in the volume of the data, the size of the graph is also increasing [1]. Graphs are well-suited for analyzing interconnections. The researchers are showing a lot of interest in using graph databases for mining data from social media [2], [3]. Graph databases are also beneficial for business analytics disciplines that involve complex relationships and dynamic schema such as identifying the source of an IP telephony issue, supply chain management, and creating "customers who bought this also looked at." Recommendations [4].

Graphs are broadly used in modeling of complex structures in several commercial and scientific applications such as social networks [5], [6], chemical compounds [7], [8], biological networks [9], etc. Analyzing a large graph with millions and billions of nodes is challenging. Recently much research have been done on large graphs [10], [11].

The graph mining tasks such as graph Partitioning, graph clustering, etc. are usually computationally exhaustive and do not scale well to very large graphs. To deal with this challenge, graph sampling provides an effectual and inexpensive solution for the large graph analysis. Graph sampling is a statistical technique for analysis used to select, analyze and manipulate a representative subset of graph in order to identify trends and patterns in the larger graph being examined.

The sampling techniques in graphs have been used for various graph applications such as frequent subgraph mining [12], graph clustering, graph classification etc. Recently, Graph sampling have also been used for anomaly detection [11] and query evaluation [13]. Chu and Sethu [14] have introduced a greedy algorithm based on spectral radius for adding vertices having high eigen-value centrality to the sample subgraphs.

Analyzing large graph at small scale, while capturing the original graph properties is convenient and relatively easy task for analysis. But, to create a representative but a small sample from a massive large graph is a challenging task. Several graph sampling algorithms were proposed in literature including Random Node sampling [15], Random Edge Sampling, Frontier Sampling [16], Random Walk Sampling [17]. However, all of them have certain drawbacks and provides inconsistent results.

In this paper, an efficient Influence based sampling algorithm named 'IS' for sampling a single large graph is proposed. Influence sampling (IS) analyze the influence of the vertices in the graph and produces sample by keeping into consideration that the important vertices of the graph should be present in the sample. The experiments are performed over three graph datasets from real life. Also, the performance of the proposed influence sampling is compared with the existing sampling algorithms such as random node sampling, random walk sampling [15] and frontier sampling [18]. It is shown that the proposed approach provides consistency in terms of accuracy and is efficient in terms of processing time in most of the cases.

The rest of the paper is structured as follows: First, the preliminaries are given in section 2. The existing sampling approaches are discussed in section 3. Section 4 introduces the proposed sampling approach influence sampling. The details about the performance metrics are given in section 5. The experimental evaluation is given in section 6. At last a precise conclusion of the proposed approach and the results is given in section 7.

II. PRELIMINARIES

A Graph G can be defined as $\{V, E\}$, where $|V|$ is a set of vertices and $E \in [V \times V]$ is a set of edges. Let the size of sample S_i be x and sampling factor be δ.

Definition 1: The sampling of graph $G = (V,E)$ is to generate a subgraph $G' = (V',E')$ where, $V' \subseteq V$ and $E' \subseteq E$.

978-1-5386-3077-8/17/$31.00 ©2017 IEEE

Definition 2: Given a graph $G=(V,E)$ the sampling factor δ can be defined as the ratio $V|/|V'|$, where $V' \subseteq V$.

For example when $\delta=5$, the subgraph G' will have 20% of the total vertices of graph G.

In the proposed algorithm, degree is considered as a parameter to select the appropriate sample. Formally degree of a vertex can be defined as:

Definition 3: The degree d_i of a vertex $v_i \in V$ in a graph G is the number of edges incident on vertex v_i.

A good sample subgraph G' of G should have similar properties such as degree distribution, clustering coefficient, etc. as of graph G.

III. Sampling Approaches

A. Random Nodes/Edges(*RN*)

Random sampling algorithms are popular and have been used extensively. Existing research shows that despite of the simplicity, random sampling produces good quality of graph samples. The random Node (*RN*) [15] produces a graph sample by first selecting a subset of vertices *v* randomly. Later the induced subgraph of the original graph is formed by selected the edges that exist between the selected vertices *v*. In Random Edges (*RE*) the graph samples are formed by random selection of edges from the original graph *G*.

However, these methods have a bottleneck. The *RN* approach may select the isolated nodes for generating samples and *RE* approach might result in the formation of sparsely connected graphs that violate the community structure of the original graph *G*.

B. Random Walk based Sampling (*RW*)

The random walk based sampling method selects an initial seed vertex *s* randomly and traverse the graph by selected the neighbours of the seed randomly. At each step, RW moves fro from the current node *s* to the adjacent node *u* of *s*. However, *RW* might get stuck in isolated component of graph.

To avoid this, a random jump is added to *RW* approach. Such that, in each step of *RW*, the walker either follows a randomly selected adjacent vertex of the current node or with probability α, it selctes a random new vertex from the graph vertices. In general α is taken as 0.15 [15].

C. Frontier Sampling (*FS*)

It is an edge sampling technique that performs *m* independent random walks on graph *G*. It starts with selecting a set of *S* vertices as seeds. The k^{th} walker of FS begin with node $S(k)0$, where $k=1,\ldots,m$. From the selected *s* seeds, a seed *v* is selected with the probability $P(v)$ given as:

$$P(v)=\frac{k_v}{\sum_{u\in S} k_v} \quad (1)$$

From the outgoing edges of v, an edge (v,u) is selected. The vertex *v* is replaced by *u* in the set of seeds *S* and the edge (v,u) is added to the sequence of sampled edges. These steps are repeated until a predefined budget is reached. The budget is often defined as the average number of vertices divided by the number of random walks [16].

IV. The Proposed Approach

The proposed approach Influence Sampling (*IS*) explore graph based on the degree of the vertices. The idea is to select the most popular vertices as seeds. Degree of the vertices is a most widely used to determine the popular vertices in the graph. Vertex with high degree is more influential than the other neighbor vertices with comparatively low degree.

The most popular *m* vertices are selected from the graph based on the degree values of the vertices. These *m* vertices are considered as seed nodes. The set of *m* seed nodes is represented by $s=\{s_1,\ldots,s_m\}$. For providing high graph coverage, it is taken into consideration that those seed nodes are selected that is at least two edges far from each other.

For each seed node s_i, the adjacent nodes of s_i are ranked based on their degree. The adjacent node of s_i with highest degree and edge (s_i,w) are included in sample S_i. Then in next iteration, the node *w* is considered as seed node and the adjacent neighbors of the new seed nodes are analyzed based on their degree. The seed nodes do not consider the previous seed node for the exploration as it is already there in the sample S_i.

For an instance, during any iteration, two adjacent nodes of the seed nodes may have the same degree. In that case, one adjacent node is selected randomly from the two nodes with high degree.

Algorithm 1: **Influence Sampling (G,m,δ)**

Input: A Graph $G(V,E)$, Number of seeds *m*, sampling Rate δ, sample size *x*

Output: A subgraph S_i of size *x*

1. Rank *N* vertices of graph based on the decreasing order of their degree d_i
2. Initialize seed vertices set *s* by selecting top vertices from the sorted degree list
3. $S_i = 0$
4. **While** $S_i < x$ do
5. *NewSeed*=0
6. **For each** s_i
7. Rank adjacent vertices of s_i based on degree values
8. Select adjacent vertex *w* with highest degree
9. **If** $s_i \rightarrow w$
10. *NewSeed*=*w*
11. **End**
12. **Else**
13. *NewSeed*=Random(s_j)
14. **End**
15. **End**
16. $S_i \leftarrow \{s_i,w\} \bigcup \{(s_i,w)\}$
17. $w=s_i$
18. **End**

Also, there may exist a situation when all the adjacent neighbors of s_i are already included in sample S_i. In that circumstance, the next seed node is selected randomly from the list of seed vertices *s*. The same procedure is then repeated by taking into notice that the vertex already in the sample is not added in the sample S_i. The details of the proposed Influence sampling are given in Algorithm 1.

The size *x* of the sample depends totally on the sampling factor δ such that $0 \leq \delta \leq 1$. The size of the sample *x* can be given as:

$$x = \frac{\delta * N}{10} \tag{2}$$

where, *N* is the number of vertices in the graph *G*. The proposed approach is giving high importance to the popular vertices of the graph. The reason behind this is that the popular vertices are the most influential in the graph. They should be present in sample for better analysis.

Moreover, for providing high graph coverage, it is taken into consideration that two seed nodes should not be adjacent to each other at beginning. As, they might get struck with the same neighbor set. Furthermore, to collect the samples of high accuracy, the vertex if revisited remains in the sample. Only the in between edges are not added in the sample. Thus, the same vertex can be visited multiple times from the different paths.

V. Performance Metrics

A. Degree Distribution

Degree is defined as the number of links incident upon a vertex. The degree can be inferred in terms of the immediate risk of a vertex for catching whatever is flowing through the network such as some information or some virus. In the case of a directed network, the two separate measures of degree centrality are defined, namely in-degree and out-degree.

Accordingly, in-degree is a count of the number of links directed to the vertex and out-degree is the number of links that the vertex directs to others. When links are concomitant to some positive aspects such as collaboration or friendship, in-degree is frequently interpreted as a form of popularity, and out-degree as gregariousness.

B. Clustering Coefficient

Another frequently used metric in graphs is Clustering Coefficient. It is a measure of the degree to which the vertices of a graph tends to form a cluster. There are two versions of this measure: Local (LCC) and Global (GCC). The LCC of a vertex compute the closeness of vertex v with its neighbors to form a clique while, the GCC detects the overall indication of the clustering in graph [19].

The local clustering coefficient for a vertex *v* in an undirected graph is given as:

$$CC_v = \frac{2\,|E(v,u)|}{d_v(d_v - 1)} \tag{3}$$

Here, d_v is the degree of vertex v and $|E(v,u)|$ is the number of edges among the neighbors of vertex *v*.

The average clustering coefficient can be calculated as:

$$Avg(CC) = \frac{1}{N}\sum_{i=1}^{N} CC_i \tag{4}$$

where, *N* is the number of vertices in graph *G*.

C. Edge Betweenness

The betweenness can be computed by finding the shortest paths between vertices *i* and *j*. Let σ_{ij} be the shortest path between vertices *i* and *j*. The betweenness centrality for a vertex *v* is the number of shortest paths σ_{ij} that pass through the vertex *v*. It is given as:

$$BC(v) = \sum_{i \neq v \neq j} \frac{\sigma_{ij}(v)}{\sigma_{ij}} \tag{5}$$

Edge betweenness can be defined as the number of shortest paths σ_{ij} in the graph *G* that pass through a given edge (*u*, *v*).

VI. Experimental Evaluation

All the experiments were carried out on i5-4570S processor having 2.90GHz clock speed. The system have 4 GB RAM running on Ubuntu 14.04 64-bit Linux operating system. Three datasets are considered for the evaluation from social networks and collaboration network. The details of each dataset are given in Table 1. All the datasets are taken from Stanford large network dataset collection [20].

Table I. Dataset Description

Dataset	Type	Number of Nodes	Number of Edges
D1	Facebook	4039	88,234
D2	Twitter	81,306	1,768,149
D3	DBLP	317,080	1,049,866

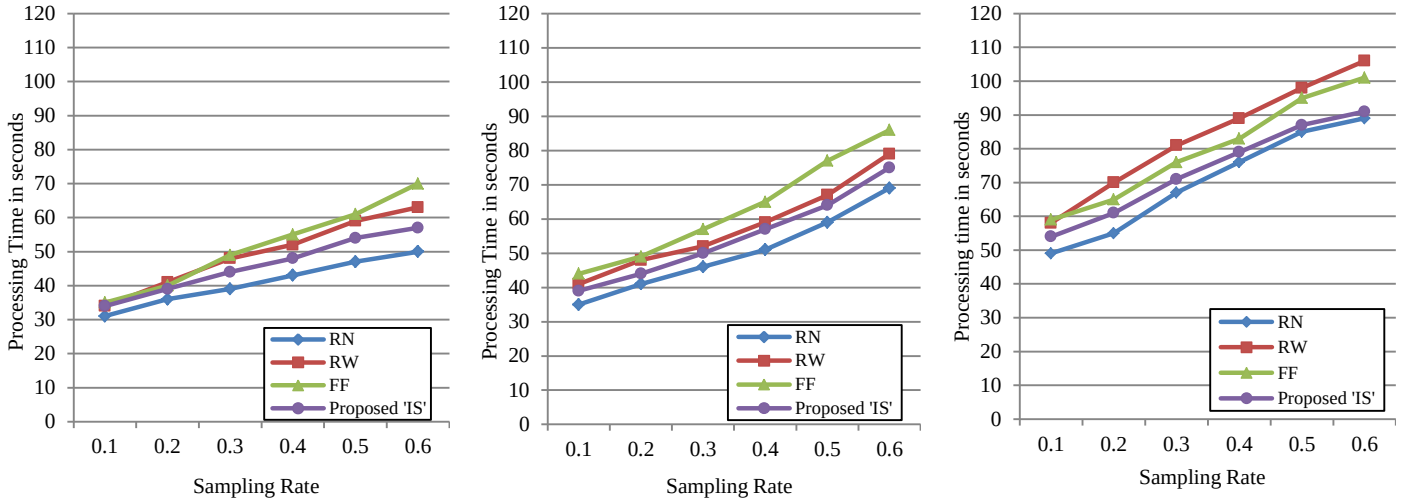
First dataset is from Facebook social network representing a friend list. Each vertex indicates a user and an edge between two users signifies the friendship between them.

Twitter is also a social network where the users follow other users. DBLP is a co-authorship network. In the dataset used, vertices are authors and their exist an edge between two authors if they publish at least one paper together.

We performed experiments over the above three real-life datasets and compared the performance of proposed Influence sampling with the state-of-art sampling algorithms such as Random Node (RN) sampling, Random Walk (RW) sampling and Frontier Sampling (FS).

We examined the characteristics of the sampled graphs by varying the sampling rate δ. Let the size of the original graph as 1, sampling rate δ=0.4 means that the size of sample is 40% of the original graph.

We compared the performance of all the algorithms on the basis of processing time in performing sampling. Further, the algorithms are compared based on quality of samples prepared in terms of Degree similarity, Clustering-Coefficient Similarity and betweenness similarity.



(a) (b) (c)

Fig. 1. Processing Time v/s Sampling Rate (a) Facebook (b) Twitter (c) DBLP

The characteristics of the sample S_i are compared with respect to the characteristics of original graph G using a metric:

$$\frac{p(S_i)}{p(G)} \times 100 \qquad (6)$$

where, $p(S_i)$ is a particular characteristic of sample S_i and $p(G)$ is the value of the same characteristic of graph G.

A. *Processing time*

We compared the processing time of the proposed Influence sampling with the other state-of-art sampling algorithms by varying the sampling rate δ. The results for the same are depicted in Fig. 1. For better analysis, we experimented for the sampling rates 0.1, 0.2, 0.3, 04, 0.5 and 0.6.

It can be noticed that for larger sampling rates, the proposed Influence sampling becomes more efficient in terms of processing time than the Random walk and Frontier sampling. However, Random Node sampling proved to be having lesser processing time for all the considered sampling rates because of its simplicity. But, for DBLP dataset, the difference between processing time of proposed Influence sampling and Random Node sampling is minimum for sampling rates 0.5 and 0.6. The proposed Influence sampling performs better then Random Walk and Frontier sampling and achieves comparable results with simple random node sampling in terms of processing time.

B. *Accuracy*

The accuracy of the proposed algorithm is measured in terms of degree distribution, clustering coefficient and betweenness for the sampling rates $\delta = \{0.1, 0.2, 0.3, 04, \text{and } 0.5\}$.

The degree similarity characteristic of sampling algorithms with the original graph is shown in Fig. 2. In Facebook dataset, for $\delta = 0.1$, the best degree similarity is generated by the proposed IS. However, for $\delta = 0.5$, Random walks produce better samples in terms of the degree distribution. But the performance of Influence sampling remains consistently efficient as shown in Fig. 2(a).

For twitter graph dataset, Influence sampling outperforms all the competent algorithms for all the sampling rates. In this case, the random node sampling performs worst of all the algorithms. However, frontier sampling performs as well as influence sampling and is defeated by influence sampling be a slight margin.

For the citation graph of DBLP, influence sampling again performs well as shown in Fig. 2(c). It can be observed from the Fig. 2 that, the random walk sampling has the most inconsistent results in terms of degree distribution and Influence sampling and frontier sampling depict high degree similarity.

Conversely, in the case of average global clustering coefficient, there is a drastic change in the performance of the considered sampling algorithms. The plot in Fig. 3 shows the average fraction of triangles present around the vertices of degree d. Instinctively, it gives an idea about the community structure in the graph. The Random node sampling surprisingly has very high clustering coefficient accuracy for the DBLP citation graph as shown in Fig. 3(c) and has low accuracy for Facebook graph as shown in Fig. 3(a) Frontier sampling also have good accuracy and provides consistent results for all the datasets.

However, the proposed influence sampling provides more consistent results for all the sampling rates and is highly accurate for all the considered datasets.

Betweeness computes the shortest paths in between the vertices of the graph. As the proposed influence sampling considers the degree of the vertices for sampling thus, has smaller diameter and better betweenness as depicted in Fig. 4. For all the datasets, Influence sampling performs better than the frontier sampling, random walk and random node sampling.

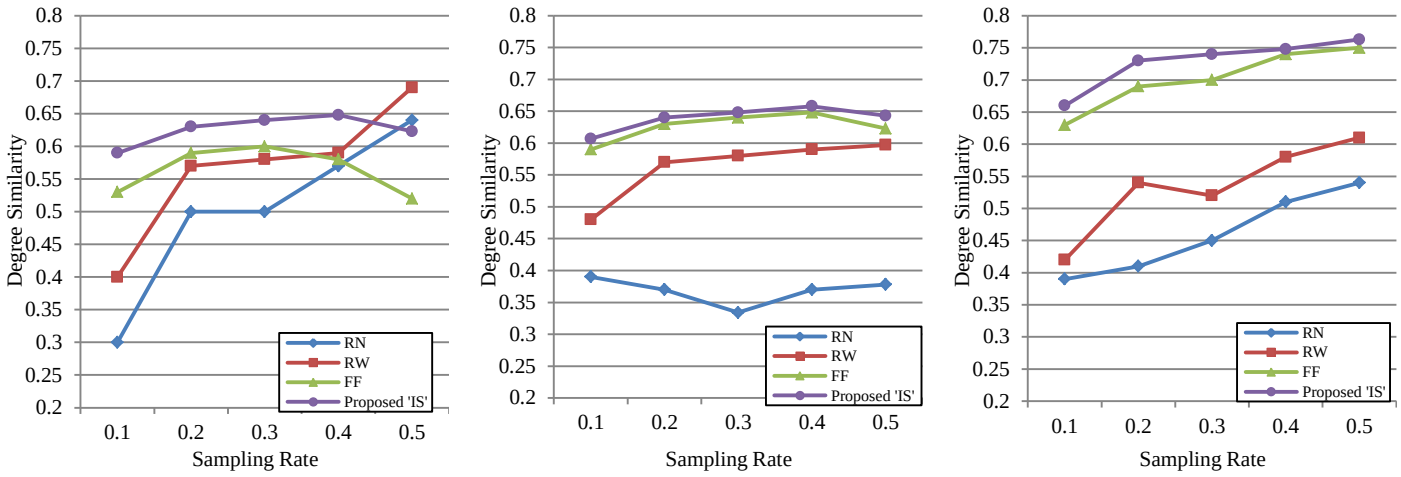


Fig. 2. Degree Distribution Similarity vs. Sampling Rate (a) Facebook (b) Twitter (c) DBLP

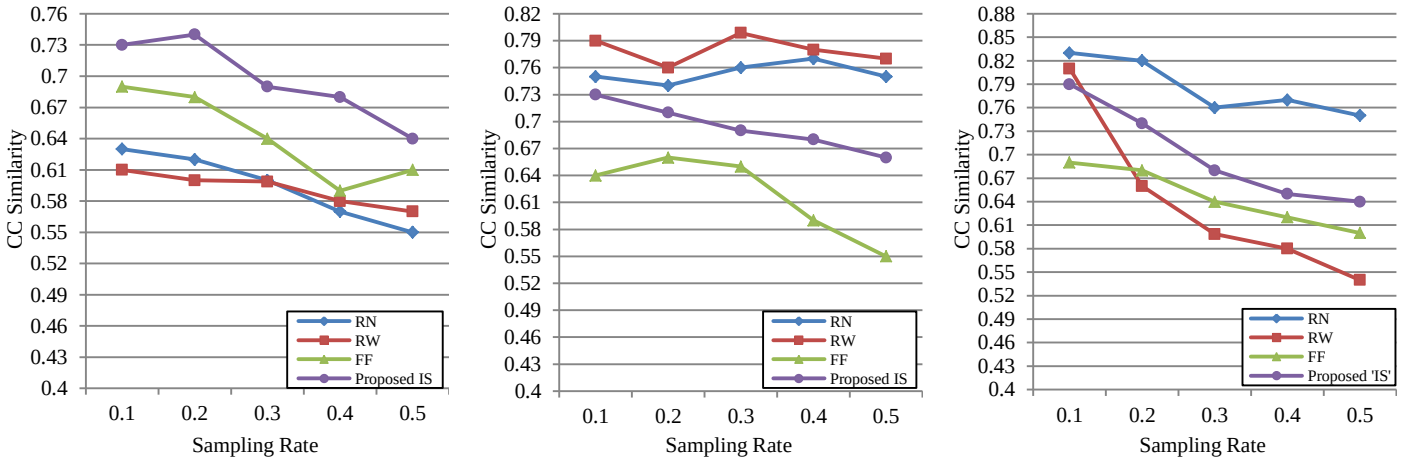


Fig. 3. Clustering Coefficient Similarity vs. Sampling Rate (a) Facebook (b) Twitter (c) DBLP

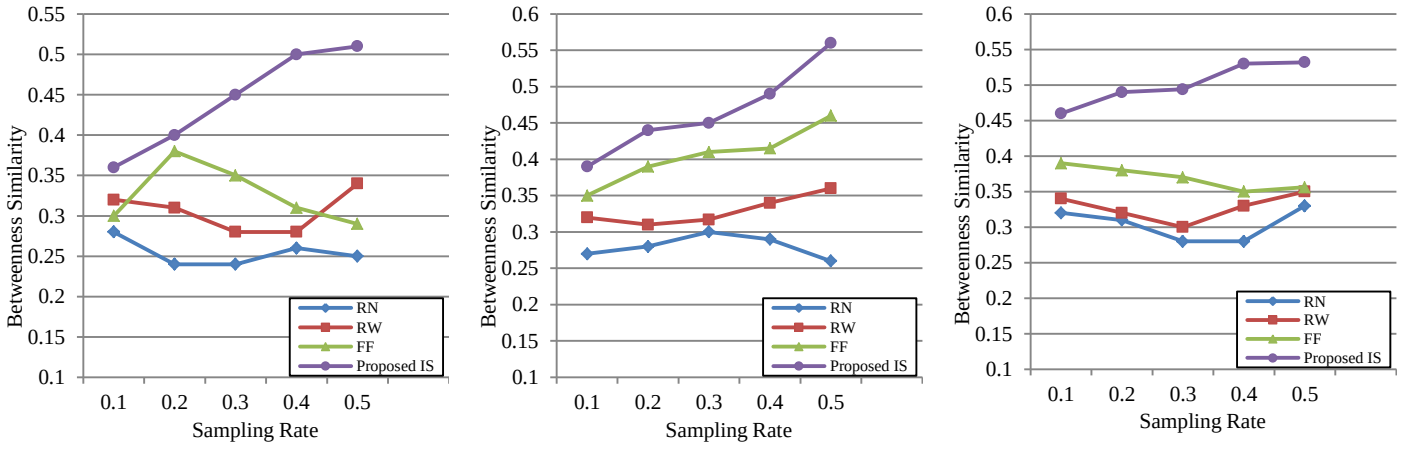


Fig. 4. Betweenness Similarity vs. Sampling Rate (a) Facebook (b) Twitter (c) DBLP

Conclusion

In this paper, an efficient algorithm named Influence sampling (IS) for sampling a large graph is presented. The proposed IS approach take into consideration the degree of the vertices for sampling in every iteration. The experimental evaluation is performed on three real life graph datasets. The accuracy is measured in terms of degree similarity, clustering coefficient similarity and betweenness similarity. It is shown that the proposed approach performs well for all the considered graph datasets. The performance of the proposed approach is compared with three existing sampling approaches random node sampling, random walk sampling and frontier sampling. The proposed Influence sampling outperforms the competent algorithms in most of the cases and provides consistency in the results for all the sampling rates.

In future, the more analysis will be done to study characteristics of samples produced by Influence sampling. The Graph coverage will also be analyzed. Further, the computational cost of the proposed approach will be minimized.

References

[1] C. C. Aggarwal, "Managing and Mining Graph Data (Advances in Database Systems)," p. 600, 2010.

[2] A. Bonato and Y. Tian, "Complex Networks and Social Networks," *Adv. Netw. Anal. its Appl.*, pp. 1–18, 2013.

[3] I. X. Y. Leung, P. Hui, P. Liò, and J. Crowcroft, "Towards real-time community detection in large networks," *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.*, vol. 79, no. 6, 2009.

[4] D. J. Cook and L. B. Holder, *Mining Graph Data*. Wiley, 2007.

[5] S. Malek, M. Golsefid, M. Hossien, and F. Zarandi, "Fuzzy Community Detection Model in Social Networks," *Int. J. Intell. Syst.*, vol. 30, pp. 1227–1244, 2015.

[6] A. Bonato *et al.*, "Dimensionality of social networks using motifs and eigenvalues," *PLoS One*, vol. 9, no. 9, pp. 1–7, 2014.

[7] M. Kuramochi and G. Karypis, "Frequent subgraph discovery," *Proc. IEEE Int. Conf. Data Min.*, pp. 313–320, 2001.

[8] M. Kuramochi and G. Karypis, "An Efficient Algorithm for Discovering Frequent Subgraphs ∗," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 9, pp. 1038–1051, 2004.

[9] K. Shi, L. Gao, and B. Wang, "Systematic tracking of coordinated differential network motifs identifies novel disease-related genes by integrating multiple data," *Neurocomputing*, vol. 206, pp. 3–12, 2016.

[10] V. Bhatia and R. Rani, "A parallel fuzzy clustering algorithm for large graphs using Pregel," *Expert Syst. Appl.*, vol. 78, 2017.

[11] C. Complexity, B. A. Miller, N. Arcolano, M. M. Wolf, and N. T. Bliss, "Spectral Anomaly Detection in Very Large Graphs."

[12] R. Zou and L. B. Holder, "Frequent subgraph mining on a single large graph using sampling techniques," *Proc. Eighth Work. Min. Learn. with Graphs - MLG '10*, pp. 171–178, 2010.

[13] R. Li, J. X. Yu, R. Mao, and T. Jin, "Recursive Stratified Sampling : A New Framework for Query Evaluation on Uncertain Graphs," vol. 28, no. 2, pp. 468–482, 2016.

[14] X. Chu and H. Sethu, "On Estimating the Spectral Radius of Large Graphs through Subgraph Sampling," no. NetSciCom, pp. 432–437, 2015.

[15] J. Leskovec and C. Faloutsos, "Sampling from large graphs," *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discov. data Min. - KDD '06*, p. 631, 2006.

[16] B. Ribeiro and D. Towsley, "Estimating and Sampling Graphs with Multidimensional Random Walks," 2010.

[17] L. Edwards, L. Johnson, M. Milosavljevic, V. Gadepally, and B. A. Miller, "Sampling Large Graphs for Anticipatory Analytics," 2015.

[18] B. Ribeiro and D. Towsley, "Estimating and Sampling Graphs with Multidimensional Random Walks."

[19] J. J. Pfeiffer and J. Neville, "Methods to Determine Node Centrality and Clustering in Graphs with Uncertain Structure," in *The Proceedings of International AAAI Conference on Web and Social Media*, 2011, pp. 1–9.

[20] J. Leskovec and K. Andrej, "Stanford Large Network Dataset Collection," 2014. [Online]. Available: https://snap.stanford.edu/data/. [Accessed: 10-Mar-2017].