LETTER

On Random Walk Based Weighted Graph Sampling

Jiajun ZHOU[†a)], ***Member***, **Bo LIU**[†], **Lu DENG**[†], **Yaofeng CHEN**[†], ***and*** **Zhefeng XIAO**[†], ***Nonmembers***

SUMMARY Graph sampling is an effective method to sample a representative subgraph from a large-scale network. Recently, researches have proven that several classical sampling methods are able to produce graph samples but do not well match the distribution of the graph properties in the original graph. On the other hand, the validation of these sampling methods and the scale of a good graph sample have not been examined on weighted graphs. In this paper, we propose the weighted graph sampling problem. We consider the proper size of a good graph sample, propose novel methods to verify the effectiveness of sampling and test several algorithms on real datasets. Most notably, we get new practical results, shedding a new insight on weighted graph sampling. We find weighted random walk performs best compared with other algorithms and a graph sample of 20% is enough for weighted graph sampling.

key words: *weighted graph sampling, graph mining, graph scale*

1. Introduction

Various types of networks such as online social networks (OSNs), peer-to-peer networks (P2P) and World Wide Web (WWW) have drawn much attention from researchers of different domains, including psychology, mathematics, social science, computer science and behavioral science. Many types of networks are formalized as graphs so that numerous researchers are familiar with graph computing and graph-based analysis. However, in some scientific and industrial domains, the scale of graph is so large that it is difficult to handle such a large graph within limited time. With the increase of the graph scale, the number of computing resources increases at the same time. It is not advisable to purchase computing resources for any data analysis problem. On the other hand, time efficiency is also a main concern in graph-based analysis problem. In most cases, the size of graph is too large to get the accurate result to meet the time requirement.

Graph sampling techniques [1], [2], [4], [6], [7], [11] spring up to handle the abovementioned problem. Recent researches mainly focus on unweighted graph, investigating an effective approach to sample graph. However, weighted graph sampling still remains unsolved with less attention because each edge in weighted graph is assigned to a numeric weight. On the other hand, weighted graph sampling is different from unweighted graph sampling. The existence of edge weight changes the graph property distribution of original graph. The unweighted graph sampling techniques are no longer suitable to sample weighted graph for the inconsistency of the graph property distribution between sample graph and original graph. Our work provides a perspective view of graph sampling techniques, introduces a weighted graph sampling problem and explore the applicability of sampling algorithms to deal with that problem.

In this paper, we consider a weighted graph sampling problem. Given a large weighted graph, our goal is to get a representative weighted sample graph that have similar attributes. Our sample graph should hold the properties of the original graph and be similar to the original graph as much as possible. To be more specific, the distance between the graph property distributions of original graph and sample graph is expect to be as small as possible.

In this work, we answer the questions that what is the scale of a good weighted graph sample and how to evaluate the performance of a weighted graph sample. We formalize the problem of weighted graph sampling. The sampled weighted graph should have similar graph properties as the original graph. We consider edge weight and node weight as the evaluation metric for the uniqueness compared with unweighted graph. The other interesting metrics such as degree and average path length are omitted in this paper due to the page limit and computation cost. We take Kolmogorov-Smirnov Test as evaluation method and evaluate several algorithms over a real weighted graph collected by other researchers [10]. We find Weighted Random Walk outperforms much than any other algorithms in all metrics introduced in this paper.

An overview of our contributions is presented as follows.

- We present a weighted graph sampling problem of sampling representative subgraph from a given large-scale graph.
- As for the weighted graph sampling problem, we give a thorough and complete analysis on several algorithms, testing the performance on real datasets, with graph properties of edge weight and node weight.
- We perform a systematic evaluation of sampling algorithms, introducing Kolmogorov-Smirnov test to characterize the similarity of graph property distribution between the original graph and the sampled graph.

Manuscript received August 14, 2017.
Manuscript revised October 3, 2017.
Manuscript publicized November 1, 2017.

[†]The authors are with College of Computer, National University of Defense Technology, Changsha, 410073, China.
a) E-mail: zhoujiajun@nudt.edu.cn (Corresponding author)

DOI: 10.1587/transinf.2017EDL8179

2. Weighted Graph Sampling

2.1 Problem Definition

Let $\mathcal{G} = <\mathcal{V}, \mathcal{E}, \mathcal{W}>$ be an undirected and weighted graph with $|\mathcal{V}| = n$ nodes and $|\mathcal{E}| = m$ edges. $\mathcal{V} = <v_i>$ is the vertex set where each v_i is a vertex in the graph. $\mathcal{E} = <e_{ij}>$ is the edge set where e_{ij} is an edge connected with vertex v_i and v_j. Each edge e_{ij} is undirected. To simplify it, we view edge e_{ij} and e_{ji} as the same edge in our problem. For each weight $w_{ij} \in \mathcal{W}$, w_{ij} is the weight assigned to the edge e_{ij}.

Given a large weighted graph, our primary goal is to obtain a representative weighted sample graph that have similar attributes through a specific sampling algorithm on an original graph. Our sample graph should hold the graph properties of the original graph and be similar to the original graph as much as possible. To be more specific, the distance between the graph property distributions of original graph and sample graph is expect to be as small as possible. The smaller the distance is, the better the sample graph is. Take edge weight as an example. A good sampled graph is that the edge weight distribution of which is similar to the edge weight distribution of original graph so that the graph properties are kept through effective sampling methods. We note that unweighted graph sampling is different from weighted graph sampling. As mentioned in [8], the edge selecting probability distribution on unweighted graph is $\pi_{unweighted} = 1/|\mathcal{E}|$. However, the edge selecting probability distribution on weighted graph is $\pi_{weighted} = w_{e_{ij}} / \sum_{e \in \mathcal{E}} w_e$. The selecting probability is thus changed for the edge weight.

2.2 Sampling Algorithms

Many researchers have concentrated on simple graph sampling to improve efficiency and obtain a more similar sample graph. However, few works pay attention to weighted graph sampling, which still remains unsolved. The performance of these simple graph sampling algorithms on weighted graph has not been validated. Weighted graph is different from simple graph for it contains edge weight. Here we describe some classical simple graph sampling algorithms and their variant.

Independent Edge Sampling (IES): Independent Edge Sampling is an intuitive method to achieve a uniform distribution of edge weights. Consider an edge set E, independent edge sampling randomly selects an edge with replacement and with probability proportional to the edge weight. The distribution of sampled edge weight from weighted graph is equivalent to the distribution of edge weight from original graph, which is

$$\pi_{e_{ij}} = \frac{w_{e_{ij}}}{\sum_{e \in \mathcal{E}} w_e} \tag{1}$$

For an unweighted graph, each edge weighted is equivalent. Then, the selecting probability of all edges from the same node are the same. It is explicit that the unweighted version is a variant of the weighted version. Thus, the distribution of sampled edge weight from unweighted graph is equivalent to the reciprocal of the edge number, denoted as

$$\pi_{e_{ij}} = \frac{1}{|\mathcal{E}|} = \frac{1}{m} \tag{2}$$

Breadth-First Search (BFS): Breadth-First Search [2] is commonly used in web search, exploring edges in a systematic way. BFS starts from the seed node and expands by the boundary between explored nodes and unexplored nodes at each iteration. Each neighbor node of current node is explored before moving to next level. A traverse tree is then generated consisting of all nodes within certain distance from the seed node.

Random Walk (RW): Random Walk [7] randomly selects a node with equal probabilities. The neighbors of the current node are all candidate nodes to be selected. Consider a node v with n neighbors and next-hop node w, the transition probability from v to w is

$$p_{vw} = \frac{1}{deg\,(v)} \tag{3}$$

Note that $deg(v)$ is the node degree of node v. As our problem only exists in undirected graph, it is no need to make distinctions between out-degree and in-degree.

Metropolis Hastings Random Walk (MHRW): Metropolis Hastings Random Walk [4] is an unbiased algorithm on unweighted graph. The algorithm takes proper transition probability so that the walk can obtain a desired uniform distribution of node degree. We view node v as current node and node w as the next node to be selected. Then, the transition probability is noted as:

$$p_{vw} = \min\left(\frac{1}{deg(v)}, \frac{1}{deg(w)}\right) \tag{4}$$

Weighted Random Walk (WRW): Weighted Random Walk [12] is a variant of random walk. Different from random walk, weighted random walk select next-hop node w from current node v with unequal probabilities. Consider an edge e_{vw} connected with node v and node w, the edge weight is defined as w_{vw}. The node weight is the sum of all the connected edge weights. For example, the node weight of node v is illustrated as w_v. The transition probability from v to w is

$$p_{vw} = \frac{w_{vw}}{w_v} = \frac{w_{vw}}{\sum_{u \in neighbor(v)} w_{vu}} \tag{5}$$

3. Experiment Evaluation

3.1 Dataset Description

We use a real world weighted graph as our experiment dataset, which is the HEP-TH network collected by M. Newman [10]. The HEP-TH network is a collaboration network scientists posting preprints on the high-energy theory archive. Table 1 summaries the basic statistics of the

Table 1 Basic statistic of HEP-TH network.

Graph properties	Value
Total Nodes	8361
Total Links	15751
Mean Degree	3.77
Mean Edge Weight	0.973
Clustering Coefficient	0.294

weighted graph used in our experiment.

3.2 Performance Metrics

To effectively evaluate the efficiency of several algorithms, we adopt Kolmogorov-Smirnov Test to examine algorithm performance metrics.

Kolmogorov-Smirnov Test. The K-S test [9] named after two Soviet mathematicians Kolmogorov and Smirnov is a nonparametric test, qualifying the distance between the empirical distribution function of the sample and the cumulative distribution function of the reference distribution. To compare the sampling result, we use K-S test to compute the vertical distance between two distributions, where F_S and F_R represent sample distribution and reference distribution. The equation is denoted as

$$KS(F_S, F_R) = max|F_S(x) - F_R(x)| \tag{6}$$

3.3 Experimental Results

Compared with unweighted graph, weighted graph is different in that edges in weighted graph are assigned with multiple labels or attributes. We use four algorithms to sample the whole graph. As mentioned in [6], 15% sample is enough to match the properties of the original unweighted graph. We sample the graph six times in each algorithm. But the scale of sampled weighted graph is to be examined, which is the sampling ratio here. The sampling ratio is the percentage of the number of sampled edges compared with the original graph. In each step, edges are selected by different sampling algorithms. The nodes connecting to the edge at both ends are also collected to form the subgraph to further examine the graph properties. We use each algorithm to sample the graph on a scale of 10% to 60%. Then, we have six different subgraph in different scale for each algorithm and four subgraph in the same sampling ratio.

To examine the performance of our algorithm, we evaluate several sampling algorithms on real datasets in terms of the distribution of edge weight and node weight. We use Independent Edge Sampling (IES) as ground truth, randomly selecting an edge with the probability proportional to its weight. The algorithm can achieve a subgraph with uniform and stationary distribution of edge weight from original graph, which we explain it in Sect. 2.2. Moreover, in some circumstance, when the entire graph is not accessible, like crawling the whole graph, IES is an effective method to examine the efficiency of crawling algorithm.

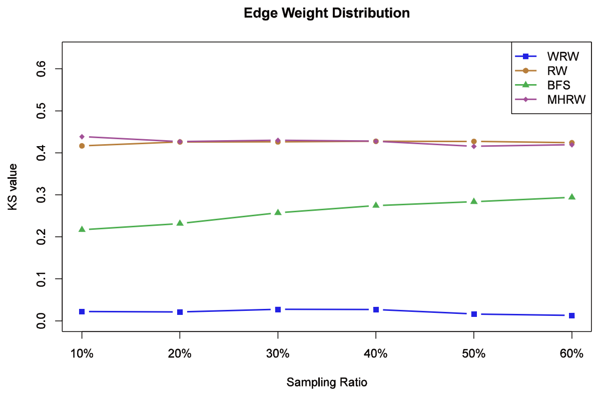

Fig. 1 Edge weight KS value

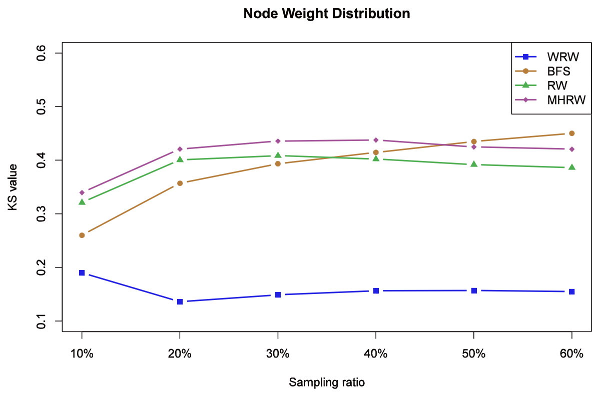

Fig. 2 K-S test

In each sampling ratio, a subgraph is obtained. We compute the distribution of graph properties from diverse subgraphs by different sampling algorithms. We compare the graph properties of the subgraphs generated by specific algorithm in the same scale with the subgraph by IES. Then, we compute edge weight distribution and node weight distribution to test the efficiency.

- *Edge weight distribution:* Figure 1 demonstrates the K-S test of edge weight distribution. For each algorithm, the K-S test value of the edge weight distribution is explicitly performed on the plot. We noted that RW performs a similar results as MHRW. The average KS values of RW and MHRW are 0.424 and 0.426 separately, reflecting that RW and MHRW are biased on weighted graphs due to the high KS value. BFS get the second lowest KS value, the mean value of which is 0.259. However, as the sampling ratio increases, the KS value of BFS increases. BFS is not a desired algorithm in our problem. In contrast, WRW performs best in compare with other sampling methods. The average KS value is 0.021. This is a remarkable result for the distance of the edge weight distribution is extremely low when compared with the ground truth.
- *Node weight distribution:* As depicted in Fig. 2, the KS value from different sampling scale by diverse sampling algorithms are plotted as lines. Similar with edge weight distribution, RW and MHRW perform the simi-

lar results, the average KS value of which are 0.385 and 0.413 respectively. The KS values are relatively high so that RW and MHRW are not suitable to our weighted graphs sampling problem. There is an obvious raise in the line of BFS. The result do not converge to a stable value but increase from 0.26 to 0.45. It is useless to obtain an un-convergent result. WRW is clearly the most accurate in preserving node weight distribution. The mean KS value from WRW is 0.157.

In summary, these results show that WRW is the most accurate algorithm in maintaining graph properties on weighted graphs. For the edge weight, the results of WRW do not reflect direct connection to the sampling ratio. For the node weight, it seems that 20% of original graph from WRW is the most suitable sampling scale for weighted graph sampling. The main reason may be that WRW considers the effect of edge weight which other algorithms neglect. Moreover, WRW selects edges with the probability corresponding to the probability distribution of edge weight. The wrong edge selecting probability is the core of poor performance.

4. Conclusion

Much efforts have been put on unweighted graph sampling techniques. However, it is important to examine the effectiveness of sampling techniques on weighted graph. When given a large-scale network, it is hard to analyze such a huge network by the limit of computing efficiency and computing resources. In this paper, we detail a weighted graph sampling problem of sampling representative subgraph from a given large-scale weighted graph. Although there are a few sampling algorithms on unweighted graph, the accuracy of these algorithms is unknown in preserving graph properties. We provide a new approach to examine the efficiency of sampling algorithms on weighted graph. We numerate a variety of sampling algorithms and evaluate them on real weighted graph. The result is interesting. Weighted Random Walk yields samples that better match the distributions of graph properties in the sampled graphs with those of the original graph. Moreover, we find that a 20% sample is enough to preserve graph properties in weighted graphs.

References

[1] A. Mislove, M. Marcon, K.P. Gummadi, P. Druschel, and B. Bhattacharjee, "Measurement and analysis of online social networks," Proceedings of the 7th ACM SIGCOMM conference on Internet measurement, San Diego, California, USA, pp.29–42, 2007.

[2] J.D. Wendt, R. Wells, R.V. Field, and S. Soundarajan, "On data collection, graph construction, and sampling in Twitter," 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp.985–992, 2016.

[3] M. Najork and J.L. Wiener, "Breadth-first crawling yields high-quality pages," in Proceedings of the 10th international conference on World Wide Web, Hong Kong, Hong Kong, pp.114–118, 2001.

[4] X. Wang, R.T.B. Ma, Y. Xu, and Z. Li, "Sampling online social networks via heterogeneous statistics," Computer Communications (INFOCOM), 2015 IEEE Conference on, pp.2587–2595, 2015.

[5] M. Gjoka, M. Kurant, C.T. Butts, and A. Markopoulou, "Practical recommendations on crawling online social networks," Selected Areas in Communications, IEEE Journal on, vol.29, no.9, pp.1872–1892, 2011.

[6] J. Leskovec and C. Faloutsos, "Sampling from large graphs," Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, Philadelphia, PA, USA, pp.631–636, 2006.

[7] J. Lu and D. Li, "Sampling online social networks by random walk," Proceedings of the First ACM International Workshop on Hot Topics on Interdisciplinary Social Networks Research, Beijing, China, pp.33–40, 2012.

[8] D. Aldous and J.A. Fill, Reversible Markov Chains and Random Walks on Graphs, 2002.

[9] A.N. Kolmogorov, "Sulla Determinazione Empirica di una Legge Distribuzione," Ist Ital Attuari, 1932.

[10] M.E.J. Newman, "The Structure of Scientific Collaboration Networks," Proceedings of the National Academy of Sciences of the United States of America, vol.98, no.2, pp.404–409, 2000.

[11] K. Cheng, "Sampling from Large Graphs with a Reservoir," Network-Based Information Systems (NBiS), 2014 17th International Conference on, pp.347–354, 2014.

[12] L. Lovász, L. Lov, and O.P. Erdos, "Random Walks on Graphs: A Survey," Combinatorics, vol.8, pp.1–46, 1993.