Advanced Science and Technology Letters
Vol.48 (CIA 2014), pp.108-119
http://dx.doi.org/10.14257/astl.2014.48.19

A New Relational Networks Sampling Algorithm Using Topologically Divided Stratums

Xiaolin Du[1], Yunming Ye[1], Yueping Li[2], and Ge Song[1]

[1] Shenzhen Key Laboratory of Internet Information Collaboration,
Shenzhen Graduate School, Harbin Institute of Technology, China
[2] ShenZhen Polytechnic, Shenzhen, China
yaoabbe@163.com

Abstract. One popular solution to deal with large-scale relational networks is to derive a representative sample from huge relational networks. We expect this sample could represent the origin relational network well so that the sampled network can be used for simulations and further analysis instead of the origin one. In this paper, we propose a network stratified sampling algorithm using topologically divided stratums for large relational networks, which can maintain the topological similarity well between sampled network and original network. In addition, we evaluate our algorithm on several well-known data sets. The experimental results show that our algorithm outperforms the previous methods.

Keywords: sampling algorithms, relational networks, stratified, topology structure

1 Introduction

In our realistic lives, social networks, such as twitter, micro-blog, MSN, Facebook, co-citation relation, credit network, etc., appear everywhere. The modern science of networks has brought significant advances in our understanding of complex systems [1]. In research, social networks are usually represented by different types of graphs. Vertices represent entities, and edges represent interactions between pairs of entities. Some graph mining techniques, such as graph visualization techniques, graph structure analyzing techniques, etc., are then employed to assist social networks analysis. However, given a large graph with millions of vertices, it is very difficult to use typical graph mining approaches to handle the entire graph directly. An essential issue is to find certain methods to accelerate the graph mining process. A popular solution is to accomplish a sub-graph, which can represent the original graph effectively such that we are able to use this sub-graph for simulations and analysis. The accomplishment of a sub-graph relies on a graph sampling process. This process aims at selecting a set of vertices and edges in a way that the resulting sub-graph obeys some general characteristics of the original graph. In this paper, we focus on developing new methods in the context of graph sampling techniques.

Generally, sampling large graph encounters three questions [2]. What is good sampling method? What is a good sample size? How do we measure the goodness of

ISSN: 2287-1233 ASTL

a single sample as well as the goodness of a whole sampling method? At present there are some state of the art sampling algorithms: Random Node (RN) sampling, Random PageRank Node (RPN) sampling, Random Degree Node (RDN)sampling, Random Edge (RE) sampling, Random Walk (RW) sampling, Random Jump sampling(RJ), Forest Fire (FF) sampling and other sampling strategies, which we will introduce briefly in section 2. For these algorithms, sample size is set by users so that users can get their ideal sampled graph. In sampling process, maintaining similar properties between sampled graph and original graph is significant as only sampled graph represents the original graph well, can we study the sampled graph instead of the original graph. It is our aim that assisting large-scale data mining by using graph sampling technique. How to evaluate whether the sampled graph and original graph have similar properties? Now there are some techniques to measure the similarity which will be introduced in section 2.

The rest of the paper is organized as follows: Section 2 presents the related works. Section 3 describes the proposed stratified sampling algorithm using topologically divided stratums (TDS). The experiment process and the experimental results will be presented in Section 4. Finally, Section 5 concludes the paper.

2 Related Works

In this section, we will introduce some network sampling algorithms and performance evaluations respectively.

2.1 Sampling Algorithms

Currently, there have been several state-of-the-art graph sampling algorithms. Conceptually, we can split these existing algorithms into three groups [2]: methods based on randomly selecting vertices, methods relying on randomly selecting edges, and exploration techniques that simulate random walks or virus propagation to find a representative sample of the vertices.

As a typical approach based on randomly selecting vertices, Random Node sampling (RN) algorithm starts by selecting a set of vertices randomly, and then a sampled graph is induced by the selected vertices. The process of Random PageRank Node sampling (RPN) lies in setting the probability of a vertex, which is selected into the sampled graph, to be proportional to its PageRank weight. The idea of Random Degree Node sampling (RDN) is that the probability of a vertex being selected is proportional to its degree.

Similarly to RN sampling, one can also select edges randomly. This process is called Random Edge (RE) sampling. We present three methods based on exploration techniques. Random Walk (RW) sampling starts at randomly picking a vertex, and then it simulates a random walk on the original graph. Random Jump (RJ) sampling is very similar to RW sampling. The only difference is that, under RJ sampling, we randomly jump to any vertex in a graph with probability $c = 0.15$. Forest Fire (FF) sampling [3] is a recursive process. First, randomly pick a seed vertex, and begin

"burning" outgoing links and the corresponding vertex. If a link gets burned, the vertex at the other endpoint has a chance to burn its own links, and so on recursively. Apart from above-mentioned methods, there are other simple sampling strategies. In particular, Krishnamurthy et al. [4] explored contraction-based methods and graph traversal based on depth and breadth first search. But none of them performed well over all.

2.2 Performance Evaluation

The sampling algorithms enable us to utilize sub-graphs with a small- scale of vertices and edges. But, how can we evaluate the performances of these algorithms? In other words, how can we evaluate the similarity between a sampled graph and its original graph? At present, researchers have designed several evaluation measures. One strategy is to compute the similarity of the distributions of the sampled graph and its original graph to indicate their similarity. The following are representatives of existing evaluation techniques:

- The degree distribution: for every degree , we count the number of vertices with degree [5];
- The distribution of sizes of weakly connected components: we count the number of weakly connected components with the same size;
- The distribution of the clustering coefficient: let vertex have neighbors, then at most edges can exist between them; let denote the fraction of these allowable edges that actually exist, the clustering coefficient is then defined as the average over all the vertices of degree [6];
- Hop-plot: the number of reachable pairs of nodes at distance or less, where is the number of hops [7];
- The distribution of the first left singular vector of the graph adjacency matrix versus the rank ;
- The distribution of singular values of the graph adjacency matrix versus the rank: spectral properties of graphs often follow a heavy-tailed distribution [8].

Among these mature sampling algorithms and evaluation techniques, one important character of graph is overlooked, which is topological structure. Topological structure can reveal the real topology relation and social relation of networks. A good sample should maintain the similar topological structure with origin network. However, there have not been any sampling algorithms focuses on the topological structure maintenance between the original network and the sampled one. That is the focus of this paper.

In this paper, we propose a sampling algorithm which can get the sample network which has similar topological structure with origin network. In addition, we evaluate our algorithm at some existed evaluation techniques on several well-known data sets. The experimental results show that our algorithm outperforms the previous methods.

3 Algorithm Description

First, we introduce the terminologies that are frequently used in this paper. Given an initial relational graph , represents the vertex set of , and represents the edge set of . Let be a sample of graph , where represents the vertex set of , and represents the edge set of .

Our motivation is as follows. Given an initial graph , which is supposed to sample, we wish to sample the vertices and edges distributing globally in in order to maintain the topology of . That is, here are some vertices lies on almost every part of . At the same time, the sampled graph also obeys the properties above well.

3.1 Sampling Model

Before describing our model, we firstly introduce some necessary terminologies. In graph theory, the distance between two vertices in a graph is the number of edges in a shortest path connecting them. This is also known as the geodesic distance [9] because it is the length of the graph geodesic between those two vertices. If there is no path connecting the two vertices, i.e., if they belong to different connected components, then conventionally the distance is defined as infinite. The diameter of a graph is the greatest distance between any pair of vertices. To find the diameter of a graph, first find the shortest path between each pair of vertices. The greatest length of any of these paths is the diameter of the graph. Fig. 1 shows the diameter of a simple graph whose diameter is 7. The red path in Fig. 1 is one of the greatest lengths of all shortest paths of any two vertices.

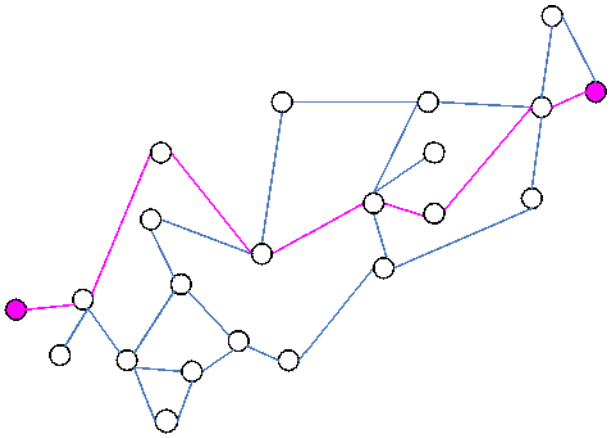

Fig. 1. Diameter of graph .

For the given graph , let denotes as the diameter of initial graph . We get two endpoints on the biggest path whose length is diameter . After randomly

picking one endpoint as start point, we add to and partition to subsets according to the distance of to , where represents the set of vertex, whose distance form is . Then sample vertices in every according to the sampling percentage. The reason we sample the vertices in using stratified strategy is that we wish the proportion of sampling vertices in every is almost the same.

We denote as the sampling vertices set of , so . Then can be split to two subsets: and . The vertex in has the property that it has at least a link to some vertex in and the vertex in has the property that it has no link to the vertices in . When sampling in , we pick percent vertices in and pick the rest vertices in . After picking vertices, we add edges to the sampling graph. Fig. 2 shows the topologically divided stratums sampling model.

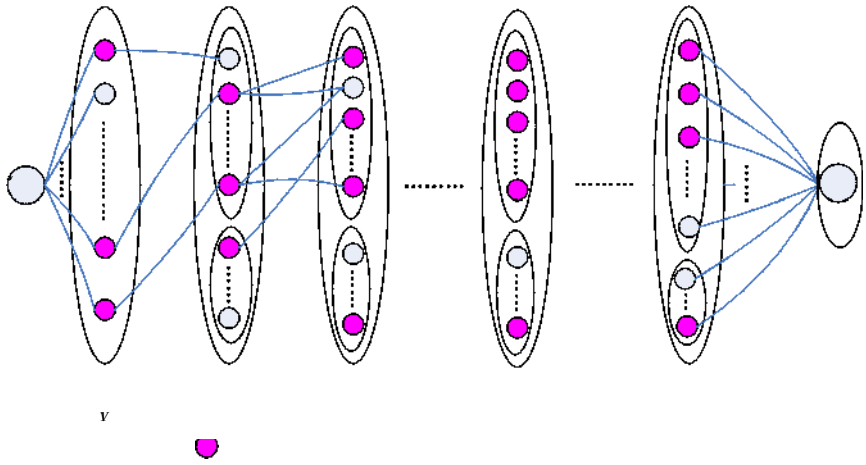

Fig. 2. Topologically divided Stratums model.

3.2 Algorithm Details

For clarity, we summarize the entire algorithm as follows. Initially, we must set two parameters: sampling percentage (or sampling size) and a percentage , whose role has been described in 3.1. Given a sampling percentage , our algorithm starts at finding two endpoints of the diameter of , and then randomly chose an endpoint as the start vertex. Algorithm 1 describes the detailed process of topologically divided stratums sampling model.

. . .

t r e e v d i t p o n n e t h e t t a r s a n a s

g

.

Algorithm 1. Algorithm Description.

Algorithm 1 Algorithm Description

Input: Original graph ;
Sampling percentage .

Output: Sampled graph .

1: Obtain the diameter of ;
2: Randomly pick one endpoint as ;
3: Split to subsets according to the distance to ;
4: Add to ;
5: ;
6: **while** **do**
7: Split to and ;
8: Randomly pick percentage vertices in to ;
9: Randomly pick percentage vertices in to ;
10: ;
11: **end while**
12: Add edges to the sampling graph ;

Thus, the sampling on topologically divided stratums model begins at obtaining the diameter of original graph , randomly choosing an endpoint of diameter as start vertex , and then splits to subsets according to the distance to , samples vertices in every recursively until all vertex stratums has been sampled. In this process an important point is that we randomly pick percent vertices in when sampling in . Such a strategy can maintain the connectedness of sampled graph. Meanwhile, sampling in every stratum can also make the "sampling" not lie in a local part of graph but disperse all over graph as our aim is to sample the vertices and edges distributing globally in in order to maintain the topology of . The sampling can be performed globally. That is, here are some vertices lies on almost every part of .

3.3 Algorithm Extensions

Our basic version of the topologically divided stratums sampling model exhibits the situation that origin graph is a connected graph. But real networks are not connected all the time, which may have numbers of connect components. By extending this model to real networks in natural ways, we propose an extension method: we can do the "stratums sampling" process in every connect component. That is, we must add an extra step, which is to get the connect components of origin graph. Then run out algorithm in every connect component. Fig. 3 shows a graph with 4 connect components, and we do the "stratums sampling" process in 4 connect components.

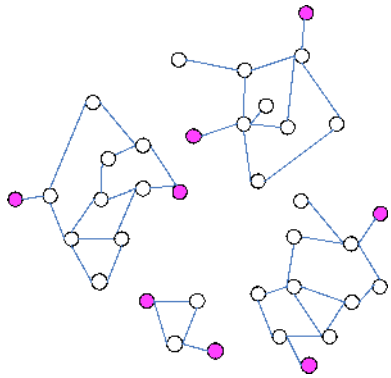

Fig. 3. Stratums sampling in every connect component.

4 Experiments

In this section we will present the experimental results on several real graphs. We consider five common used datasets coming from the homepage of Newman [10]. They are Email, Power, Hep-th, astro-ph and cond-mat. Hep-th, astro-ph and cond-mat these three datasets are not connected, so we get their biggest weak connected component and denote them "hep-th_conect", "astro-ph_connect" and "cond-mat_connect" respectively. The following table is the detail description of these five datasets. Table 1 detailed describes the five data sets and Fig. 4 shows the visualization layouts of five data sets.

In statistics, the Kolmogorov-Smirnov test (K-S test) is a nonparametric test for the equality of continuous, one-dimensional probability distributions that can be used to compare a sample with a reference probability distribution (one-sample K-S test), or to compare two samples (two-sample K-S test) [11]. The smaller of test value the larger of probability that two samples obey same distribution. Thus, we employ K-S test to measure the similarity of two distributions in our paper.

Next we evaluate our algorithm and present the results in Table 2 and Table 3. Each entry in the table is obtained by averaging the K-S Test over 20 runs per dataset. These 2 tables show the experimental results by 2 different sampling percentages (P). For each column we bold the best test value. In result tables we can observe our algorithm gets most of the best test values.

For dataset "cond-mat_connect" and dataset "astro-ph_connect", our algorithm covers almost the best test value for all sampling percentages. The size of these two datasets is larger than the other three datasets, and our algorithm performs better when the scale of networks rose. For dataset "power", our method cannot get the best performance. Fig. 4 (c) is the layout of "power", and from Fig. 4 we can observer that the distribution of "power" differs from the other datasets. The "power"'s diameter is 46 which is larger than the others, and vertices in "power" are not distributed radially around some centroid, but dispersed irregularly. So our algorithm does not suit for this kind of datasets. Along with the sampling percentage increase, the test values of

our method in all five datasets have tend to decrease, as more samples of original graph can represent the original structure better. From the analysis above, we can conclude our method is better than the others.

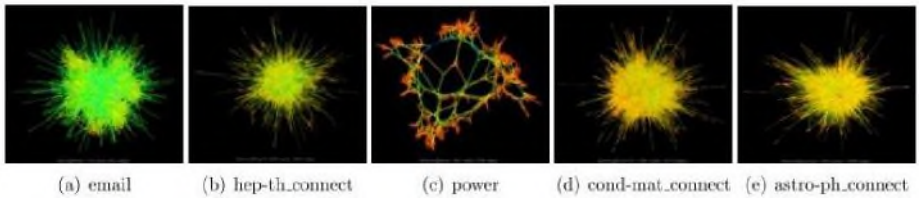

Fig. 4. Datasets description.

Table 1. Datasets description.

Dataset Name	Edge Count	Vertex Count	Diameter	Description
email	1134	5452	8	List of edges of the network of e-mail interchanges between members of the University Rovira i Virgili (Tarragona) [12].
hep-th_conect	13815	5835	19	Weighted network of coauthor ships between scientists posting preprints on the High-Energy Theory E-Print Archive between Jan 1, 1995 and December 31, 1999[13].
power	6594	4941	46	An undirected, unweighted network representing the topology of the Western States Power Grid of the United States [14].
cond-mat_connect	44619	36458	18	Weighted network of coauthor ships between scientists posting preprints on the Condensed Matter E-Print Archive between Jan 1, 1995 and December 31, 1999[13].
astro-ph_connect	119652	14845	14	Weighted network of coauthor ships between scientists posting preprints on the Astrophysics E-Print Archive between Jan 1, 1995 and December 31, 1999[13].

Table 2. Statistic results of 5 datasets on 3 evaluation criteria. P=0.1.

P=0.1	Data Sets Name	email	hep-th_connect	power	cond-mat_connect	astro-ph_connect
RN	CD	0.9856	0.9636	0.9922	0.8327	0.7573
	Degree	0.5314	0.5155	0.8674	0.4718	0.3783
	Hop_plot	0.0541	0.0706	0.0226	0.0392	0.0888
RDN	CD	0.9535	0.9611	0.9815	0.9888	0.8084
	Degree	0.4339	0.6769	0.8031	0.7690	0.3471
	Hop_plot	0.0431	0.0444	0.0275	0.0600	0.0615
RPN	CD	0.9384	0.9729	0.9862	0.9611	0.9875
	Degree	0.5728	0.6482	0.7949	0.4611	0.3366
	Hop_plot	0.0375	0.0311	0.0242	0.0277	0.0742
RE	CD	0.9887	0.8957	0.9852	0.8774	0.5427
	Degree	0.4457	0.6741	0.8532	0.7924	0.4920
	Hop_plot	0.0325	0.0676	0.0222	0.0353	0.0570

RW	CD	0.9989	0.9999	0.9996	0.9413	0.9707
	Degree	0.7068	0.6601	**0.4480**	0.4487	0.1068
	Hop_plot	0.0174	0.0018	0.0018	0.0026	0.0028
RJ	CD	0.9993	0.9882	0.9974	0.9956	0.9715
	Degree	0.7603	0.7879	0.4732	0.6383	0.2156
	Hop_plot	0.0043	3.70E-05	3.66E-09	1.79E-04	9.33E-04
TDS	CD	**0.8012**	**0.7510**	0.9999	**0.2635**	**0.1200**
	Degree	0.7012	0.7082	0.7034	**0.1109**	**0.0799**
	Hop_plot	**2.32E-04**	**2.35E-04**	**9.23E-11**	**7.98E-06**	**2.32E-04**

Table 3. Statistic results of 5 datasets on 3 evaluation criteria. P=0.2.

P=0.2	DataSets Name	email	hep-th_connect	power	cond-mat_connect	astro-ph_connect
RN	CD	0.9563	0.9670	0.9687	0.7908	0.7010
	Degree	**0.4138**	0.5939	0.7566	0.5017	0.4176
	Hop_plot	0.0297	0.0776	0.2792	0.0559	0.0767
RDN	CD	0.9949	0.9514	0.9325	0.9960	0.9732
	Degree	0.4928	0.6905	0.8303	0.7593	0.3604
	Hop_plot	0.0202	0.0530	0.0300	0.0813	0.1012
RPN	CD	0.9840	0.9879	0.9347	0.9966	0.9997
	Degree	0.5263	0.6499	**0.7368**	0.6171	0.2342
	Hop_plot	0.0116	0.0283	0.0298	0.0415	0.0563
RE	CD	0.9997	0.8962	**0.9131**	0.9887	0.6302
	Degree	0.5198	0.7081	0.8531	0.8618	0.6071
	Hop_plot	0.0743	0.0404	0.0271	0.0628	0.0797
RW	CD	1.0000	0.9999	1.0000	0.9471	0.9412
	Degree	0.7606	0.8341	0.8050	0.3845	0.0146
	Hop_plot	0.0059	0.0035	5.07E-04	0.0013	0.0056
RJ	CD	0.9928	0.9991	0.9990	0.9951	0.9338
	Degree	0.8108	0.7642	0.8239	0.5879	0.1234
	Hop_plot	0.0011	0.0007	1.31E-07	8.77E-04	**0.0055**
TDS	CD	**0.7456**	**0.7917**	0.9957	**0.1329**	**0.0020**
	Degree	0.7523	**0.3078**	0.8597	**0.0896**	**0.0017**
	Hop_plot	**0.0015**	**2.56E-04**	**1.64E-10**	**1.89E-04**	0.0079

Fig. 5 and Fig. 6 plot the scatter results of experiments. Form these figures we observe our algorithm covers the most of best results again 2 sampling percentage.

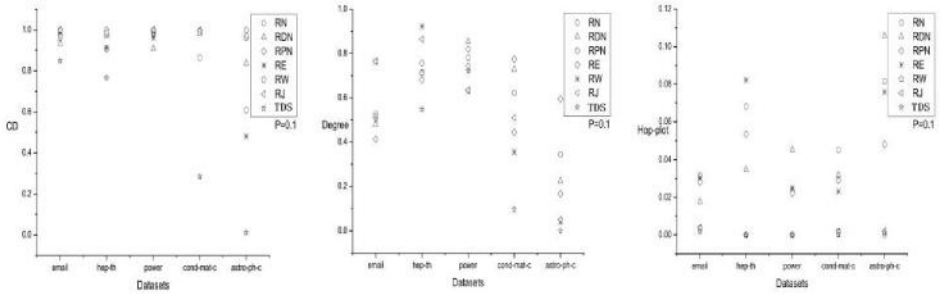

Fig. 5. Evaluation result scatter plots by p = 0.1.

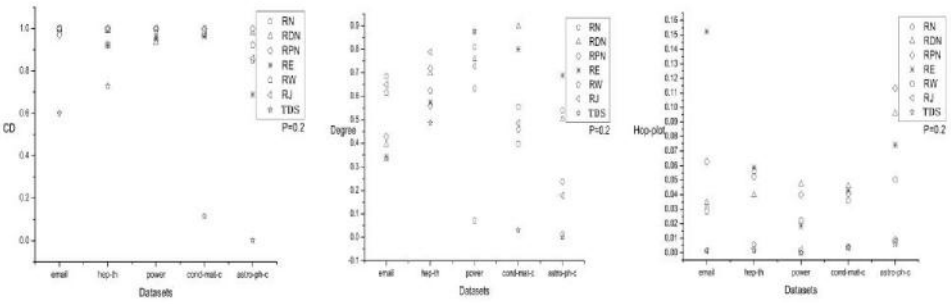

Fig. 6. Evaluation result scatter plots by p = 0.2.

Fig.7, Fig.8, Fig.9, Fig.10 and Fig.11 show the contrastive layout results of 7 sampling algorithms on 5 datasets. From the results we get the message that algorithms base on randomly choosing nodes or edges can induced so much unexpected isolated vertices in simple graphs and failed in maintaining similar topological structure between original graph and sample graph. Algorithms based on exploration can maintain this similarity better. After comparing these visualization results and origin graphs, we conclude that our algorithm performs better than the other algorithms based on exploration (RW, RJ).

Fig. 7. Visualization results of “email” dataset.

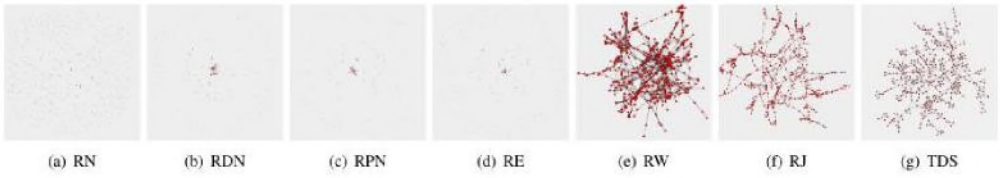

Fig. 8. Visualization results of "hep-th_connect" dataset.

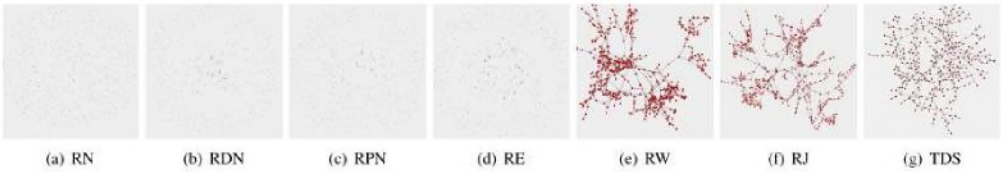

Fig. 9. Visualization results of "power" dataset.

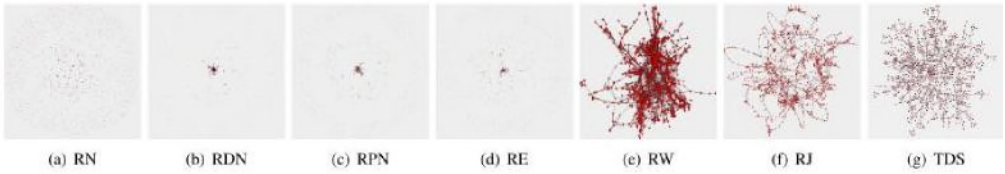

Fig. 10. Visualization results of "cond-mat_connect" dataset.

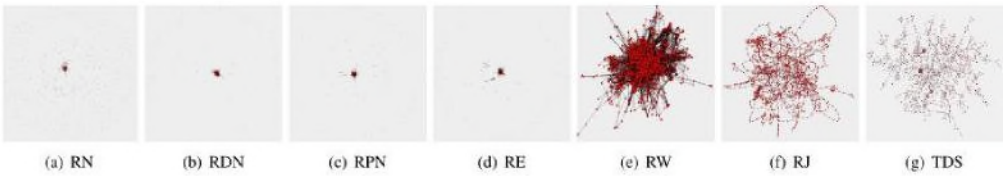

Fig. 11. Visualization results of "astro-ph_connect" dataset.

5 Conclusions

It is very significant that generating a representative sampled graph when predigesting large scale graph to accelerate the process of graph mining. Despite there are some existed evaluations and algorithms about sampling graphs, there has been relatively little work on the properties of topological similarity between original graph and sampled graph. This is exactly the focus of this work. The main findings and contributions in this paper is that we propose a topologically divided stratums sampling algorithm based on only two parameters. We provide thorough analysis and comparison in the published literature, testing multiple sampling algorithms (6), on several datasets (5), with 3 graph evaluation methods. We perform a systematic evaluation of sampling algorithms by non-trivial statistical evaluation methods (the

Kolmogorov-Smirnov Test). Then we conclude our algorithm can capture evaluations observed both in previous works and maintain the topological similarity between original graph and sampled graph.

Reference

1. Fortunato, S.: Community detection in graphs, Physics Reports, vo.486, pp.75-174, 2010.
2. Leskovec J., Faloutsos C.: Sampling from large graphs. KDD'06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 631-636, 2006.
3. Leskovec J., Kleinberg J., Faloutsos C., "Graphs over time: Densification laws, shrinking diameters and possible explanations," ACM SIGKDD, 2005.
4. Krishnamurthy V., Faloutsos M., Chrobak M., Lao L., Cui J.-H. and Percus A. G., "Reducing large internet topologies for faster simulations," In Networking, 2005.
5. Faloutsos M., Faloutsos P. and Faloutsos C., "On power-law relationships of the internet topology," In SIGCOMM, pp.251-262, 1999.
6. Watts J. D., Strogatz H.S., "Collective dynamics of 'small-world' networks," Nature, vo.393, pp. 440-442, 1998.
7. Palmer C. R., Gibbons P. B. and Faloutsos C., "Anf: A fast and scalable tool for data mining in massive graphs," In SIGKDD, 2002.
8. Chakrabarti D., Zhan Y. and Faloutsos C., "R-mat: A recursive model for graph mining," In SDM, 2004.
9. Bouttier J, Francesco PD , Guitter E. Geodesic distance in planar graphs. Nuclear Physics B 2003; 663(3):535–567.
10. http://www-personal.umich.edu/~mejn/netdata/
11. http://en.wikipedia.org/wiki/KS_Test#cite_note-0
12. Guimera R., Danon L., Diaz-Guilera A., Giralt F. and Arenas A., "Self-similar community structure in a network of human interactions," Physical Review E, vo.68: 065103(R), 2003.
13. Newman M., "The structure of scientific collaboration networks," Proceedings of the National Academy of Sciences of the United States of America, vo.98, pp.404-409, 2001.
14. Watts J. D., Strogatz H.S., "Collective dynamics of 'small-world' networks," Nature, vo.393, pp. 440-442, 1998.
15. Stumpf M.P.H., Wiuf C., May R.M., "Subnets of scale-free networks are not scale-free: Sampling properties of networks," In PNAS, vo.102, 2005.
16. Ruoyu Zou and Lawrence B. Holder, "Frequent Subgraph Mining on a Single Large Graph Using Sampling Techniques," In Proc. 8th Wshop. Mining and Learning with Graphs (MLG 10), 2010.
17. Wang T, Chen Y, Zhang Z, et al. "Understanding graph sampling algorithms for social network analysis," Distributed Computing Systems Workshops (ICDCSW), 31st International Conference on. IEEE, pp.123-128, 2011.
18. Zou R, Holder L B. "Frequent subgraph mining on a single large graph using sampling techniques," Proceedings of the Eighth Workshop on Mining and Learning with Graphs. ACM, pp.171-178, 2010.

