Graph Property Preservation under Community-Based Sampling

Ruohan Gao, Pili Hu, Wing Cheong Lau
Department of Information Engineering, The Chinese University of Hong Kong
{gr013, hupili, wclau}@ie.cuhk.edu.hk

***Abstract*—With the explosion of graph scale of social networks, it becomes increasingly impractical to study the original large graph directly. Being able to derive a representative sample of the original graph, graph sampling provides an efficient solution for social network analysis. We expect this sample could preserve some important graph properties and represent the original graph well. If one algorithm relies on the preserved properties, we can expect that it gives similar output on the original graph and the sampled graph. This leads to a systematic way to accelerate a class of graph algorithms. Our work is based on the idea of stratified sampling [14], a widely used technique in statistics. We propose a heuristic approach to achieve efficient graph sampling based on community structure of social networks. With the aid of ground-truth of communities available in social networks, we find out that sampling from communities preserves community-related graph properties very well. The experimental results show that our framework improves the performance of traditional graph sampling algorithms and therefore, is an effective method of graph sampling.**

***Index Terms*—CBS sampling, graph property preservation, graph algorithm acceleration**

I. Introduction

Many properties have been defined to characterize a graph and these properties are very important for people to understand the graph [6]. Given a large graph with millions or even billions of vertices and edges, it is very difficult to use typical graph mining approaches to handle the original graph directly. As a result, various graph sampling techniques have been proposed for the analysis or mining of large-scale complex networks. To our understanding, the incentive of doing graph sampling is to get a small transformed graph from the original large graph with preserved properties. If so, running the algorithm on the transformed graph has approximately the same effect as running it on the original graph. Moreover, we can estimate the properties of the original graph using the transformed graph. There will be no point if the graph transformation procedure takes even longer time than straightforward computation on the original graph.

What we want to propose is a sampling method, which creates a sub-graph that well preserves graph properties. In the meantime, it should be efficient and simple. Random Node sampling and Random Edge sampling, as representatives of the most classical graph sampling methods, are indeed efficient and simple. However, the results of applying these two algorithms on the original graph directly are less than satisfactory [8]. It occurs to us that what if we apply these state of the art sampling algorithms in a different way?

Community structure has become one of the most important topological structure properties of complex networks. In real-life social networks, nodes explicitly join various social groups based on shared interests or background. Such groups can be used to define a reliable and robust notion of ground-truth communities. Yang and Leskovec [19] comprehensively studied a set of 230 large real-world social, collaboration and information networks and defined network communities based on ground-truth. For instance, students from the same school, fans of a pop star and customers who purchase the same product can all be regarded as a community. Note that we do not need to do community detection to get these communities, because the community information is readily available from the ground-truth. As such, graph pre-processing time or complexity should not be the concern of our framework. Therefore, we propose a new graph sampling method based on the ground-truth of community structure and classical graph sampling algorithms. If we do graph sampling in every community independently and then combine the sampling results of each community. We can expect that the final resultant graph should reflect the community-related graph properties very well.

The rest of the paper is organized as follows: Section II introduces the related work, which mainly describes common sampling objectives and approaches; In Section III, we propose our Community-Based Sampling (CBS) framework in detail; Section IV introduces the experiments and our analysis of the experimental results; a brief conclusion is given in Section V.

II. Related Work

Many graph sampling algorithms have been proposed and all these algorithms have a sense of randomly selecting vertices or edges (maybe according to current knowledge of the graph). However, they arise from different contexts and have different problem dimensions or foci. In this section, we provide a short taxonomy of graph sampling works. Refer to [6] for a more detailed survey.

A. Common Sampling Objectives

1) Get a subset of representative vertices: This is the usual motivation from sociology studies, e.g. poll the opinion of the sampled vertices (people) [16]. In many scenarios, target population can be sampled directly, e.g. phone number, random street survey, etc. In other scenarios, target population is hidden, e.g. drug abusers in urban area. In this latter case,

researchers have to execute certain graph sampling algorithm on a graph to explore the hidden population.

2) Preserve certain property of the original graph: A property of a graph can be viewed as a (possibly vector) function $f(G)$. Sometimes, we pursue exact property preservation. Sometimes, we only want to preserve the property within certain error margin. After performing sampling, two things can be done:

- Estimate graph properties: If we know some property is preserved on the sampled graph G_s, we can calculate $f(G_s)$ as an estimator for $f(G)$.
- Support graph algorithms: Many graph algorithms aim at optimizing certain objective associated with some graph properties. If we can preserve those properties on G_s, we may expect to obtain similar results by running the algorithm on G_s instead of G. This gives a general method to accelerate a class of graph algorithms.

3) Generate random graph: Graph generation is an important topic in its own right. However, some works in the literature on graph generation also use the phrase like "graph sampling". One can view a graph generation model as a family of graphs $\mathcal{G}$. The generation procedure is to sample one graph G from $\mathcal{G}$. For example, the process of performing an edge sampling is indeed the generation of an Erdős-Rényi Network [2].

B. Common Sampling Approaches

1) Vertex Sampling: Vertex Sampling is the most obvious way to create a sampled graph. We first select $V_s \subset V$ directly without topology information (e.g. uniformly or according to some distribution on V). Then we let $E_s = \{(u, v) \in E | u \in V_s, v \in V_s\}$, namely only edges between sampled vertices are kept. Similar category has been defined by Leskovec in [8] where it is called "sampling by random node selection".

2) Edge Sampling: Similar to Vertex Sampling, one can also select edges from the original graph. We first select $E_s \subset E$ somehow. Then we let $V_s^{(1)} = \{u, v | (u, v) \in E_s\}$. This definition only arises in some theoretical discussions of basic sampling methods on graphs. A more realistic definition is to let $V_s^{(2)} = V$. Then the setting is the same as graph (edge) sparsification. We adopt the second definition of Edge Sampling in our work. Similar category has been defined by Leskovec in [8] where it is called "sampling by random edge selection".

3) Traversal Based Sampling: Traversal Based Sampling has a very long history and is still the research focus in recent years. It is also called topology based sampling [1] or sampling by exploration [8]. The sampler starts with a set of initial vertices (and/or edges) and expands the sample based on current observations. Doerr and Blenn [4] formalized the framework for three intuitive graph traversal methods, which are Breadth First Sampling, Depth First Sampling and Random First Sampling. Snowball Sampling [5], which is very similar to Breadth First Sampling, has long been used in sociology studies, where an investigation is performed on the hidden population (e.g. drug abusers). Forest Fire Sampling is a probabilistic version of Snow Ball Sampling and it was originally proposed in [9] as a graph generation model, which was subsequently adapted to perform graph sampling in [8]. Random Walk Sampling also arises from different context. It can be shown in [10] that Random Walk Sampling on undirected graphs results in uniform distribution on edges. Metropolis-Hastings Random Walk Sampling algorithm [11] was widely used in Markov Monte Carlo Chain to obtain a desired vertex distribution from an arbitrary undirected connected graph. Multi-Dimensional Random Walk Sampling [15], which is also called Frontier Sampling, was proposed to address the issue that the original Random Walk Sampling may have high bias according to different initial vertex(es).

Recently, [21] proposed a novel method to derive an auxiliary graph and an affiliation graph to help the graph mining process of the original target graph. The so-called hybrid social-affiliation network can help to sample a graph indirectly but efficiently. The community membership information discussed in our work can also be transformed into a hybrid social-affiliation network.

III. Community-Based Sampling

In this section, we propose the Community-Based Sampling (CBS) framework.

A. Motivation for Community-Based Sampling

Although many algorithms have been developed, a large chunk of previous literature is devoted to property estimation. That is, the sampling algorithm does not necessarily preserve certain property. As long as an accurate estimator can be developed based on the sampled graph, the sampling procedure is deemed useful. In our context, we focus on property preservation because it yields a generic method to accelerate a class of graph algorithms. This calls for a comprehensive re-evaluation of classic simple/fast sampling procedures regarding the property preservation performance. Note that, many previous sampling algorithms only utilize the graph topology (i.e. vertices and edges). On many modern social networks, one can further get some auxiliary information in form of node level attributes. The auxiliary information can be leveraged to improve the sampling procedures. Among different auxiliary information, community membership is widely available (e.g. group/school on Facebook, conference/journal in a publication network). Towards this end, we devote this paper to investigate how community membership information can be leveraged to improve existing sampling processes.

B. Overall Design of CBS

For those classical graph sampling algorithms, the sampling process is performed on the entire graph. The sampling process may result in large variance for those community-related graph metrics. It is likely that skewed number of members are sampled from each community, e.g. a large community only gets a few representatives or a small community gets a lot of representatives in the sampled graph. One usual approach to alleviate sampling variance across different types of targets is

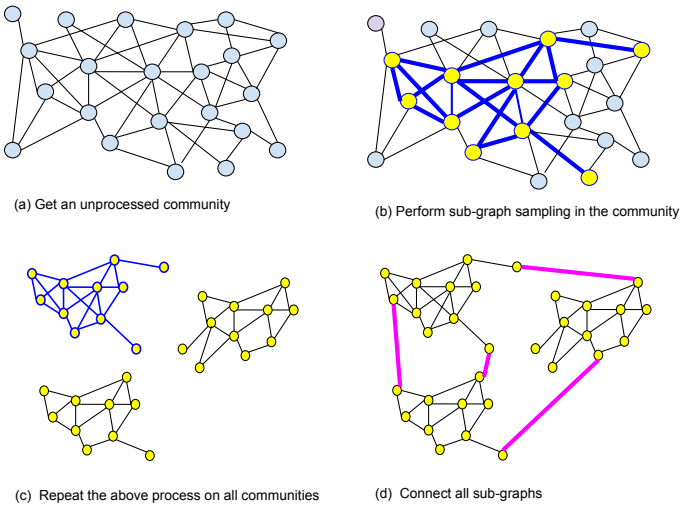

Figure 1: Sampling Process of CBS Framework

to adopt the idea of stratum sampling. The core idea of CBS framework is to use community as stratum. It first performs a sampling algorithm within each community and then connect the communities by common edges incident to those communities. Since there are two types of objects, i.e. vertex and edge, in a graph, the final result is not just stratum-based sampling on vertices. Given the different sampling algorithms used as subroutine in CBS framework, the result varies. Nevertheless, the intuition is similar to stratum sampling.

C. Procedure of the CBS Framework

The procedure of the CBS framework is illustrated in Fig 1. There are four key steps:

(a) *Get an unprocessed community*: Based on the community ground-truth obtained from online social networks, we get an unprocessed community and compose the sub-graph in the community.

(b) *Perform sub-graph sampling in the community*: We use some classical graph sampling algorithm and perform sub-graph sampling inside the community to select a certain fraction of nodes along with a set of edges.

(c) *Repeat the above process on all communities*: We perform sub-graph sampling independently inside each community to get a sampled graph for every community.

(d) *Connect all sub-graphs*: We connect all sampled graphs obtained from sub-graph sampling by adding all inter-community edges.

IV. Experimental Evaluation

In this section, we present the results of experiments on four real-life graphs to demonstrate the effectiveness of our proposed method by applying the proposed framework on eight classical graph sampling algorithms.

A. Dataset

We evaluated the proposed CBS framework on four real-world datasets.

1) DBLP collaboration network [19]: The DBLP collaboration network is a co-authorship network where two authors are connected if they co-author at least one paper. Publication venue, e.g, journal or conference, defines an individual ground-truth community; authors who published in a particular journal or conference form a community. The original dataset does not provide complete community ground-truth. Therefore, we did some pre-possessing to compose a new graph based on the communities provided as our research subject.

2) Amazon product co-purchasing network [19]: The Amazon dataset is collected by crawling the Amazon website. It is based on the "*Customers Who Bought This Item Also Bought*" feature of the Amazon website. If a product i is frequently co-purchased with product j, the graph contains an undirected edge from i to j. Each product category provided by Amazon defines each ground-truth community. Similar to the DBLP dataset, the original dataset does not provide complete community ground-truth. Therefore, we did some pre-possessing to compose a new graph based on the communities provided as our research subject.

3) Flickr photo sharing dataset [20]: Flickr is a photo sharing platform, where users can share their contents, upload tags and subscribe to different interest groups. The friendship and the commentship (i.e., who comments on whose photos) among the set of users define the ground-truth communities. The community ground-truth for this dataset is complete, and 195 communities are defined.

4) BlogCatalog social blog dataset [20]: BlogCatalog is a social blog directory website, which manages the bloggers and their blogs. This dataset is crawled from BlogCatalog which contains the friendship network crawled and group memberships. The group memberships define the ground-truth communities. The community ground-truth for this dataset is complete, and 39 communities are defined.

Dataset Name	Nodes	Edges	Communities
DBLP	260,691	949,360	13,431
Amazon	318,725	878,069	269,540
Flickr	80,513	5,899,882	195
BlogCatalog	10,312	333,983	39

B. Baselines

As mentioned in Section I, graph sampling should be efficient in order to make it a meaningful process. Therefore, the eight graph sampling algorithms we study in our work are all relatively simple and efficient classic algorithms, which have all been summarized by Leskovec and Faloutos in [8].

- *Random Node (RN) Sampling*: we uniformly select a set of nodes N at random. Then we include all the edges among the N sampled nodes, and the resultant graph is the desired sample of the original graph.
- *Random Edge (RE) Sampling*: we uniformly select a set of edges E at random. Then we include all the nodes, which serve as the endpoints of the sampled edges in E, and the resultant graph is the desired sample of the original graph.

- *Random Walk (RW) Sampling*: we uniformly choose an initial node u at random, then the next node v is randomly chosen from all of u's neighbor nodes. Node v and edge (u, v) will be sampled in this process and the random walk continues from node v. At every step, with probability $c = 0.15$ (the value commonly used in literature) we jump back to the starting node and re-start the random walk.
- *Random Jump (RJ) Sampling*: this is the same as *Random Walk* sampling, except that at every step with probability $c = 0.15$ we can randomly jump to *any* node in the graph instead of the starting node only.
- *Random Degree Node (RDN) Sampling*: we select a set of nodes N based on the degree of the nodes. The probability of a node being selected is proportional to its degree. Then we include all the edges among the N sampled nodes, and the resultant graph is the desired sample of the original graph.
- *Random PageRank Node (RPN) Sampling*: we select a set of nodes N based on the PageRank score of the nodes. The probability of a node being selected is proportional to its PageRank weight. Then we include all the edges among the N sampled nodes, and the resultant graph is the desired sample of the original graph.
- *Random Node-Edge (RNE) Sampling*: we first uniformly select a node at random and then uniformly at random choose an edge incident to the node. We repeat the above process to get the sampled graph.
- *Hybrid Random Edge (HYB) Sampling*: we perform a step of RNE sampling with probability p, or perform a step of RE sampling with probability $1 - p$. Based on the recommendation from [7], we use $p = 0.8$ in our evaluation.

C. A Case Study via Visualization

Firstly, we examine how well the sampled graph reflects the properties of the original graph by visualizing two commonly studied distributions.

- *Degree Distribution*: If we randomly choose a node $X \in V$, and let

$$p_{deg}(k) = Pr\{d(X) = k\}$$

  $p_{deg}(k)$ is thus the p.d.f. for the degree distribution. In the figure of Degree Distribution, the x-axis represents the degree of nodes, and the y-axis represents the cumulative percent, which denotes the percentage of nodes that are below the corresponding degree level. Detailed study of degree distribution has been conducted in [17].
- *Clustering Coefficient Distribution*: The local clustering coefficient of a graph is the measure of the extent to which one's friends are also friends of each other. This measure was first discussed in detail by Watts and Strogatz in a 1998 paper in Nature [18]. For node v_i, the local clustering coefficient denoted by c_i is defined as the ratio of the number of (v_j, v_i, v_k) triangles to the number of (v_j, v_i, v_k) connected triplets. Formally,

$$c_i = \frac{2l_i}{d_i(d_i - 1)}$$

  where d_i donotes the degree of node v_i and l_i denotes the number of edges between neighbors of v_i, with $c_i \in [0, 1]$. In the case where $d_i = 1$ or $d_i = 0$, we have $c_i = 0$. In the figures of Clustering Coefficient Distribution, we divide the possible value range of clustering coefficient, namely 0 to 1, into small intervals. We count the number of nodes that fall into a certain interval that we define and then we normalize the count. In the figures of this distribution, the x-axis represents the clustering coefficient, and the y-axis represents the cumulative percent, which denotes the percentage of nodes that are below the corresponding clustering coefficient level.

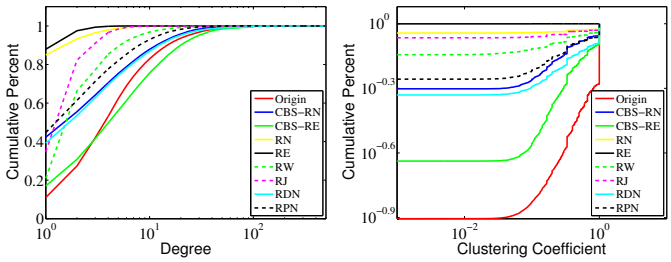

Figure 2: Distribution Visualization

For all the sampling algorithms, we use the sampling rate of approximately 10%. The evaluation is based on visual observation on how similar the two distributions of the sampled graph are to that of the original graph. For better observation, we only plot two algorithms under our CBS framework, namely CBS-RN and CBS-RE. The distributions of the original graph are obtained in advance.

Fig 2 depicts the Degree Distribution, Clustering Coefficient Distribution of the original graph and sampled graphs obtained for DBLP Dataset. We can see that the Degree Distributions and Clustering Coefficient Distributions of the sampled graphs created by CBS-RN and CBS-RE are obviously more similar to that of the original graph. The performance becomes much better under the CBS framework for Random Node Sampling and Random Edge Sampling. Because RDN sample the graph according to the degree distribution, nodes of higher degree have better chance to be sampled and these nodes are very important nodes to preserve these two distributions. Naturally, as is shown in the result, RDN also performs quite well in preserving these two distributions. Therefore, the visual inspection suggests that the proposed framework helps to preserve these two important distributions in graph sampling.

D. Graph Property Preservation

In this section, we present experimental results of all graph properties studied in our work at sampling rate of 10% to see how graph properties are preserved. For the two distributions of the previous section, we use Kolmogorov-Smirnov

D-statistic to quantify the results for comparison. K-S D-statistic can be used to compare the difference between two distributions. It is defined as $D = max_x\{|F^{'}(x) - F(x)|\}$, where x is over the range of the random variable, and F and $F^{'}$ are the two empirical cumulative distribution functions of the data. We use it in our work to compute the maximum difference between the cumulative distribution functions of the distributions obtained from the original graph and that of the sampled graph. The possible value of K-S D-statistic is between 0 and 1. The smaller the value, the more similar the property of the sampled graph is to the original graph. We also study five other global graph properties:

- *Assortativity*: Assortativity is generally defined as the Pearson Correlation of similarity between neighboring vertices. One can use degree as the similarity measure [12] and leads to the following definition. The distribution of remaining edges (except for the one that link the two vertices under consideration) is: $q_k = \frac{(k+1)p_{deg}(k+1)}{\sum_j jp_j}$. Define the joint distribution of the degree of two vertices by $e_{j,k}$. Then the assortativity is defined as:

$$r = \frac{1}{\sigma_q^2}\sum_{j,k} jk(e_{j,k} - q_j q_k)$$

  The network is said to be assortative when $r > 0$, non-assortative when $r = 0$, and disassortative when $r < 0$.
- *Average Node Degree*: Average degree of a graph is the expected value:

$$E[d(X)] = \sum_k kp_{deg}(k)$$

  where $p_{deg}(k)$ is the probability mass function of the node degree of the graph.
- *Graph Density*: Given an undirected graph $G = (V, E)$ with n nodes and m edges, graph density is defined as the ratio of observed number of edges over the maximum possible number of edges:

$$density = \frac{m}{\binom{n}{2}} = \frac{2m}{n(n-1)}$$

- *Power Law Exponent*: After obtaining the degree distribution, one can fit the power law exponent γ, s.t. the fitted distribution $p_{fit}(k) \propto k^{-\gamma}$ is closest to the observed degree distribution $p_{deg}(k)$. In [13], the detailed method of power law fitting is described and we use the following formula to extract the exponent:

$$\gamma = 1 + n[\sum_{i=1}^{n} ln\frac{d_i}{d_{min}}]^{-1}$$

  where d_i $(i = 1 \ ... \ n)$ are the measured values of degree of nodes and d_{min} is the minimum value of degree for which the power-law behavior holds. This property only makes sense when the original and sampled graphs are close to power-law graphs.
- *Average Clustering Coefficient*: The network average clustering coefficient [3], denoted by c_l, is defined as:

$$c_l = \frac{1}{n}\sum_{i=1}^{n} c_i$$

  where c_i is the local clustering coefficient of every vertex in the network.

Table I shows the results for the DBLP dataset. The last column is the ground-truth for the dataset. We can see that the CBS framework performs very well in preserving community-related graph properties like average clustering coefficient, power law exponent, degree distribution and clustering coefficient distribution. Metric values obtained under the CBS framework are generally closer to the ground-truth. However, for density and average degree, CBS does not help to preserve them since they are not quite related to the community structure of the graph. Observe that the metric values of density and average degree are much higher in sampled graphs under the CBS framework. The reason for the over-estimation is that higher degree nodes may exist in several communities at the same time in the community ground-truth. Therefore, these nodes have better chance to be sampled, thus increasing the metric value of the average degree and density in the sampled graph. The results of the other three datasets are similar, so we do not present them in the paper.

E. Comparison of algorithms with and without using CBS

In this section, we directly compare sampling results for sampling algorithms with and without CBS. We study the graph property at different sampling rate from 0% to 50%, above which graph sampling may not be quite meaningful anymore. We present our results of the two most representative properties.

Fig 3 shows the experimental results of K-S D-statistic of degree distribution. In the eight sub-figures, the blue line represents the original sampling algorithm and the green line represents the corresponding community-based sampling algorithm. Observe that, except for RDN, community-based sampling improves the performance of the original random sampling algorithm significantly. The reason why CBS does not improve the performance of RDN may be that the degree distribution in each community may not agree with the degree distribution on the global scale. Therefore, there will be no point in using RDN inside each community. Moreover, RDN sampling itself already performs quite well in preserving degree distribution.

Fig 4 shows the experimental results regarding power law exponent. Again, in the eight sub-figures, the blue line represents the original sampling algorithm and the green line represents the corresponding community-based sampling algorithm in each of the subfigure. The black dotted line represents the power law exponent for the entire network, namely the ground-truth. We can see that CBS helps the metric value to converge much faster, especially for RN, RE, RJ, RNE and HYB. The power law exponent of the sampled graphs under the CBS

Table I: Graph Properties of DBLP Collaboration Network

	RN	CBS-RN	RE	CBS-RE	RW	CBS-RW	RJ	CBS-RJ	RDN	CBS-RDN	RPN	CBS-RPN	RNE	CBS-RNE	HYB	CBS-HYB	Origin
Assortativity	0.224	0.247	0.110	0.598	0.063	0.307	-0.034	0.334	0.775	0.394	0.299	0.292	-0.046	0.319	-0.019	0.324	0.260
Average Degree	0.721	4.804	1.155	8.757	2.995	8.239	1.979	7.228	5.501	7.067	3.588	6.209	1.071	6.013	1.082	6.319	7.283
Density($\times 10^{-5}$)	2.77	16.7	4.39	35.9	11.5	31.7	7.33	27.6	20.5	28.9	13.5	25.1	3.97	22.9	4.01	24.4	2.79
Powerlaw Exponent	1.443	1.279	1.751	1.259	1.294	1.271	1.576	1.276	1.259	1.269	1.302	1.275	2.044	1.281	2.044	1.279	1.230
Average CC	0.071	0.242	$<$0.001	0.406	0.157	0.439	0.085	0.425	0.313	0.312	0.230	0.267	$<$0.001	0.329	$<$0.001	0.338	0.646
DD D-statistic	0.593	0.361	0.636	0.251	0.427	0.317	0.622	0.338	0.273	0.316	0.436	0.343	0.647	0.362	0.647	0.350	0
CCD D-statistic	0.873	0.422	0.947	0.171	0.648	0.223	0.911	0.295	0.238	0.246	0.502	0.367	0.949	0.393	0.950	0.357	0

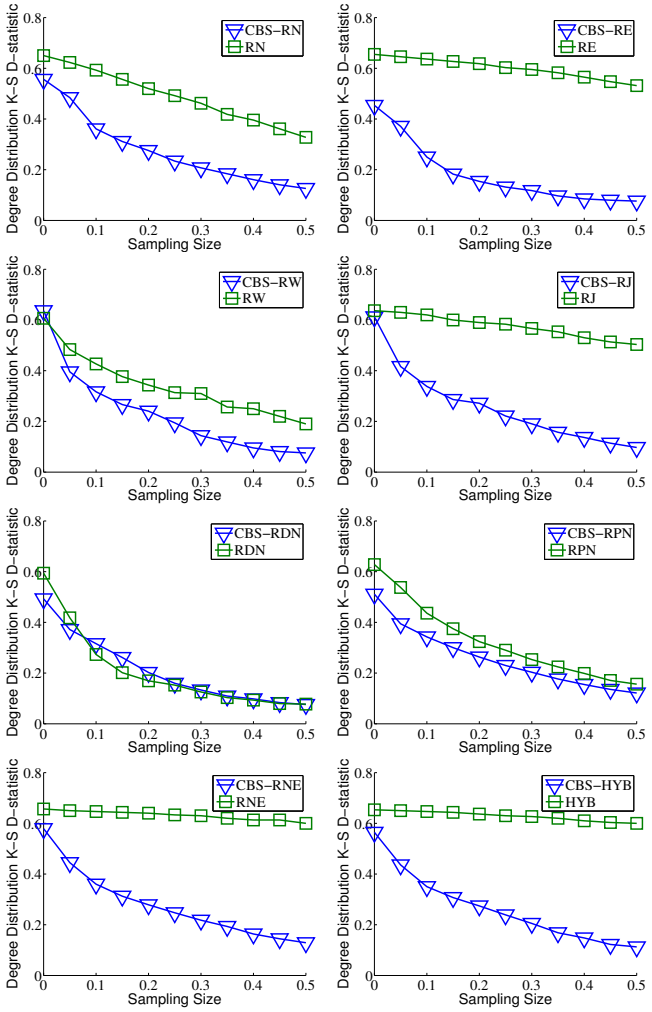

Figure 3: Degree Distribution D-statistic

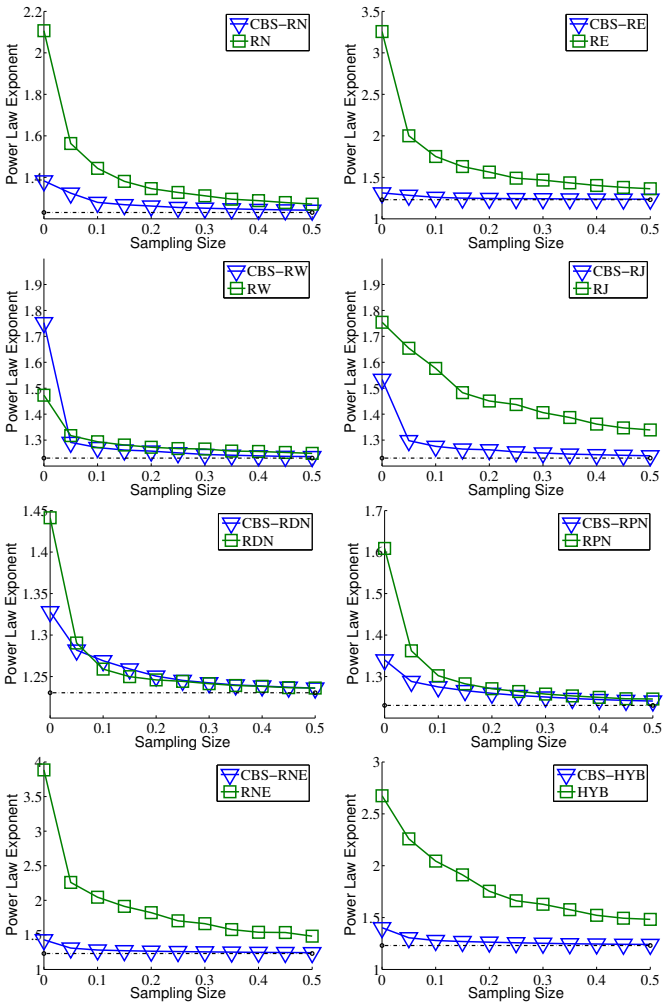

Figure 4: Power Law Exponent

framework are also very close to the ground-truth even for sampling rate as low as 5%.

F. Evaluation of Sampling Complexity

In this subsection, we present the runtime of all the graph sampling algorithms studied at sampling rate 10%. All algorithms are implemented in Python and executed on a 64 bit MacOS machine with an 2.3GHz Intel Core i7 processor and 8 GB RAM. Two input files include the original graph file in adjacency list format and the community ground-truth file, in which each line represents a community. All time measurements are average end-to-end time in unit of seconds.

From Table II, we can see that for the original algorithms which are very efficient to begin with, adding the CBS framework only slightly increases the total execution time because the same algorithm has to be applied community by community. For some algorithms like RJ, RNE and HYB, the CBS framework even reduces the execution time significantly. The reason is that it is faster to collect enough number of nodes inside each community than from the original large graph.

G. Accelerate Graph Algorithms via Sampling

One of the applications of graph sampling is to accelerate a class of algorithms by applying those algorithms on the sampled graph instead of the original large graph. Finally, we apply one algorithm on the original graph and sampled graphs

Table II: Graph Sampling Time in Seconds (Sampling Rate: 10%)

	RN	CBS-RN	RE	CBS-RE	RW	CBS-RW	RJ	CBS-RJ	RDN	CBS-RDN	RPN	CBS-RPN	RNE	CBS-RNE	HYB	CBS-HYB
DBLP	2.8	3.2	5.1	10.1	5.1	14.5	70.6	12.7	18.5	27.6	19.6	32.2	207	12.3	302.5	19.8
Amazon	2.8	4.5	5.3	20.4	24.8	29.1	78.2	34.2	23.7	231.2	28.2	278.2	248.2	34.9	367.2	51.8
Flickr	15.1	16.3	33.3	37.2	13.6	34.6	20.8	27.9	20.5	33.8	26.2	39.8	34.5	31.2	424.2	56.8
BlogCatalog	1.1	1.1	2.1	1.7	1.1	1.8	1.1	1.8	1.3	1.7	1.9	1.9	1.4	1.7	4.5	2.1

Table III: Intersection among Top 100 Ranked Nodes

	RN	CBS-RN	RE	CBS-RE	RW	CBS-RW	RJ	CBS-RJ	RDN	CBS-RDN	RPN	CBS-RPN	RNE	CBS-RNE	HYB	CBS-HYB
DBLP	7.8	41.2	7.6	40.2	15	38.6	7.2	40.7	42.7	47.1	42.3	48.3	17.2	44.3	7.2	43.1
Amazon	6.1	29.5	10.6	40.4	27.1	27.4	26.0	39.2	46.6	47.4	39.4	45.1	4.4	44.4	3.2	40.4
Flickr	8.8	20.3	9.4	50.4	20.1	58.6	18.2	57.6	56.2	59.7	63.3	59.7	10.2	61.7	9.9	62.3
BlogCatalog	10.6	12.6	35.8	69.9	52.1	70.5	42.1	72.8	79.7	72.1	80.9	76.6	50.2	68.2	58.3	69.6

to demonstrate the advantage of performing graph sampling.

In particular, we apply the PageRank algorithm on the original graph and the sampled graph to get the top 100 ranked nodes. PageRank is a way of measuring the importance of nodes and the purpose of using PageRank score is mainly to get top ranked nodes. When the original graph is very large, it is inefficient to apply the algorithm on the original graph directly. If we can get an acceptable number of top ranked nodes by applying the PageRank algorithm on the sampled graph, it will save a lot of time.

We apply the same PageRank algorithm on the original graph and the sampled graphs. To compare the results, we calculate the number of same nodes among the 100 top ranked nodes. Table III shows the corresponding results in terms of their average. We can see that algorithms under the CBS framework can generally get much more highly ranked nodes than the original algorithms. The improvement for RDN and RPN is not significant since RDN and RPN have already performed very well without the CBS framework, which is expected. RDN sample more high degree nodes, which usually turn out to rank high under PageRank. As for RPN, it samples nodes according to the PageRank score, which naturally leads to the preservation of top ranked nodes.

V. Conclusion

In this work, based on the idea of stratified sampling, we have proposed the so-called Community-Based Sampling framework. By leveraging the knowledge of ground-truth communities provided by social networks and integrate them with classical random sampling algorithms, we can sample every community independently first and then combine the subgraphs obtained, instead of sampling the original large graph directly. Through a series of experiments using real world datasets, we have demonstrated the effectiveness of CBS. Our results show that the sampled graph created by the proposed framework preserves community-related graph properties very well. It improves the performance of classical efficient graph sampling algorithms significantly without sacrificing their simplicity and efficiency. Therefore, Community-Based Sampling can serve as an effective method for performing large graph sampling.

References

[1] N. Ahmed, J. Neville, and R. R. Kompella. Network sampling via edge-based node selection with graph induction. 2011.

[2] B. Bollobás. *Random graphs*. Springer, 1998.

[3] L. d. F. Costa, F. A. Rodrigues, G. Travieso, and P. R. Villas Boas. Characterization of complex networks: A survey of measurements. *Advances in Physics*, 56(1):167–242, 2007.

[4] C. Doerr and N. Blenn. Metric convergence in social network sampling. In *Proceedings of the 5th ACM workshop on HotPlanet*, pages 45–50. ACM, 2013.

[5] L. A. Goodman. Snowball sampling. *The annals of mathematical statistics*, pages 148–170, 1961.

[6] P. Hu and W. C. Lau. A survey and taxonomy of graph sampling. *arXiv preprint arXiv:1308.5865*, 2013.

[7] V. Krishnamurthy, M. Faloutsos, M. Chrobak, L. Lao, J.-H. Cui, and A. G. Percus. Reducing large internet topologies for faster simulations. In *NETWORKING 2005. Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications Systems*, pages 328–341. Springer, 2005.

[8] 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.

[9] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 177–187. ACM, 2005.

[10] L. Lovász. Random walks on graphs: A survey. *Combinatorics, Paul erdos is eighty*, 2(1):1–46, 1993.

[11] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

[12] M. E. Newman. Assortative mixing in networks. *Physical review letters*, 89(20):208701, 2002.

[13] M. E. Newman. Power laws, pareto distributions and zipf's law. *Contemporary physics*, 46(5):323–351, 2005.

[14] J. Neyman. On the two different aspects of the representative method: the method of stratified sampling and the method of purposive selection. *Journal of the Royal Statistical Society*, pages 558–625, 1934.

[15] B. Ribeiro and D. Towsley. Estimating and sampling graphs with multidimensional random walks. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, pages 390–403. ACM, 2010.

[16] M. J. Salganik and D. D. Heckathorn. Sampling and estimation in hidden populations using respondent-driven sampling. *Sociological methodology*, 34(1):193–240, 2004.

[17] M. P. Stumpf and C. Wiuf. Sampling properties of random graphs: the degree distribution. *Physical Review E*, 72(3):036118, 2005.

[18] D. J. Watts and S. H. Strogatz. Collective dynamics of small-world networks. *nature*, 393(6684):440–442, 1998.

[19] J. Yang and J. Leskovec. Defining and evaluating network communities based on ground-truth. *Knowledge and Information Systems*, 42(1):181–213, 2015.

[20] R. Zafarani and H. Liu. Social computing data repository at asu. *School of Computing, Informatics and Decision Systems Engineering, Arizona State University*, 2009.

[21] J. Zhao, J. Lui, D. Towsley, P. Wang, and X. Guan. A tale of three graphs: Sampling design on hybrid social-affiliation networks. In *IEEE 31th International Conference on Data Engineering (ICDE)*, 2015.