Accelerating Graph Mining Algorithms via Uniform Random Edge Sampling

Ruohan Gao, Huanle Xu, Pili Hu, Wing Cheong Lau
Department of Information Engineering, The Chinese University of Hong Kong
{gr013, xh112, hupili, wclau}@ie.cuhk.edu.hk

***Abstract*—The seminal works by Karger [12], [13] have shown that one can use Uniform Random Edge (URE) sampling to generate a graph skeleton which accurately approximates all cut-values in the original graph with high probability under some specific assumptions. As such, the random subgraphs resulted from URE sampling can often be used as substitutes for the original graphs in cut/flow-related graph-optimization problems [13]. In this paper, we extend the results of Karger to show that, besides the value (weight) of the cut-set, the weights of four additional types of edge-set, namely, Volume, Association, Complement Volume and Complement Association, are all well-preserved under URE sampling. More importantly, we show that these well-preserved edge-set metrics have dominant impact on the outcome of common graph-mining tasks including PageRank computation and Community Detection. As a result, URE sampling can be used to accelerate the corresponding graph-mining algorithms with small approximation errors. Via extensive experiments with large-scale graphs in practice, we demonstrate that URE sampling can achieve over 90% accuracy for PageRank computation and Modularity-based Community Detection by sampling only 20% edges of the original graph.**

***Index Terms*—URE sampling, graph property preservation, PageRank, Community Detection**

I. Introduction

The ever-increasing popularity of Online Social Networks (OSNs) such as Facebook, Twitter, and LinkedIn has drawn a lot of attentions from researchers in recent years. Not only do these OSNs provide people with a platform to socialize, they also create ample valuable information for data mining. By applying various graph mining algorithms on these social networks, we can gain several useful insights and business intelligence. However, due to the explosive growth of the scale of OSNs in the recent years, a typical social network graph associated with these platforms easily have millions or even billions of vertices and edges. It has therefore become practically infeasible to conduct standard graph mining tasks on the original graph directly. As a result, various graph sampling techniques have been proposed for the analysis or mining of large-scale complex networks. Under such an approach, only a small subset of the nodes and/or edges from the original graph are selected to form a subgraph for further processing.

The value of any particular graph sampling scheme is heavily contingent on its ability to preserve relevant properties and metrics of the original graph, which may have dominant impact on the outcome of the corresponding graph mining algorithms. Take any Max-Flow Min-Cut algorithm as an example where the objective is to find a cut-set with minimum weight in the original graph. Here, the "total weight of a cut" can be viewed as the key graph property of interest. As long as this property can be well-preserved upon graph sampling, the result produced from the sampled graph should serve as a good approximated solution for the original graph. A key advantage of the sampled graph compared to the original one is that the former has far fewer edges (and/or vertices) and thus the graph mining algorithms can run much faster on it. Of course, the graph sampling scheme itself must be light-weight and efficient in order to yield overall computational savings when compared to the case without sampling.

The seminal results by David Karger [12], [13] establish a theoretical guarantee for preserving the cut-values of a graph via Uniform Random Edge (URE) sampling under a relatively strong condition, namely, the minimum node-degree of the original graph should be no less than $\Omega(\ln n)$ where n is the number of nodes before sampling. In this paper, we extend Karger's result to derive a more general graph-property-preservation framework. More specifically, we show that the weights of other four types of edge-set, namely, Volume, Association, Complement Volume and Complement Association, are also preserved under URE sampling if the same condition holds. Moreover, URE sampling also leads to the preservation of some other graph-theoretic properties including Ratio Cut, Normalized Cut, Ratio Association, and Normalized Association.

One major drawback of Karger's result is that real-world social graphs often cannot satisfy its required condition on minimum node-degree. Nevertheless, we will demonstrate via experiments that relavant properties and metrics can still be preserved after graph sampling. More importantly, we investigate two common graphs mining tasks, namely, PageRank computation and Community Detection on various real-world graphs. Our results show that, via URE-sampling, these graph mining tasks can yield well-approximated solutions for the original graph with substantial reduction in computation time when compared with the case of no sampling. In summary, we have made the following technical contributions:

- After reviewing the related work in Section III, we extend Karger's result on cut to other four edge-set metrics under URE sampling in Section III and provide a theoretical guarantee for graph property preservation as well.
- In Section IV, we conduct extensive experiments to quantify the extent of graph property preservation when

URE sampling is applied to large-scale social graphs in practice.

- Before concluding our work in Section VI, we, in Section V, apply URE-sampling on two graph mining tasks including PageRank computation and Community Detection to demonstrate its effectiveness in speeding up the corresponding algorithms while maintaining an acceptable level of accuracy.

II. Related Work

Researchers have proposed a lot of graph sampling algorithms in the literature. Hu et al. provide a comprehensive review for the related work in [10]. In particular, Uniform Random Edge Sampling (URE) has been extensively studied. URE Sampling scans parts or full of a whole graph and takes each scanned edge into the sampled graph with a constant probability. On the other hand, Non-Uniform Random Edge Sampling (NURE) samples edges with different probabilities. Edges with sparser connectivities are usually sampled with higher probabilities [4], [5], [9], [14]. In general, NURE Sampling can lead to a good representative of the original graph but it takes a much longer time than URE approach to obtain a sampled graph. For example, the Benczur-Karger algorithm takes $O(m \log^2 n)$ time to construct a sampled graph when the original graph is unweighted and takes $O(m \log^3 n)$ time when the original graph is weighted [4]. As a comparison, URE Sampling takes at most $O(m)$ time to construct a desired graph.

Our work focuses on graph property preservation, which is one of the most common graph sampling objectives. The concept of property preservation under large-scale graph sampling was first introduced by Leskovec and Faloutsos in [17]. They test a number of different sampling methods to assess the ability of these algorithms on preserving some important properties of the original network such as clustering coefficient, degree distribution, the distribution of component size, etc. They observe that different graph sampling algorithms are better at preserving some specific graph properties but not others, and that there does not exist one particular algorithm that can outperform all other sampling algorithms in all aspects.

Since different post-sampling applications rely on graph properties in different ways, graph sampling procedures are often tailored to specific applications. For example, Jia et al. propose several novel graph sampling methods in [11] to get a sampled graph for better visualization of large-scale power-law graphs; Krishnamurthy et al. develop methods to sample a small realistic graph from a huge Internet topology for the purpose of efficient simulation [16]; Ahmed et al. investigate the impact of network sampling on estimates of relational classification performance [3]; Chakrabarti et al. present algorithms in [8] to generate subgraphs that well preserve pairwise relationships, which are vital for applications like clustering, classification, and ranking; some other works propose to employ graph sampling approaches to address community detection problems including [19], [26] and [27].

Recently, Zhao et al. propose a novel method to derive an auxiliary graph and an affiliation graph in [28] to help the graph mining process of the original target graph. With the existence of the auxiliary graph and affiliation graph, the unbiased estimation of certain graph characteristics can be conducted efficiently. Wang et al. present Uniform Vertex Sampling and Random Walk techniques to characterize user pair properties including neighboring pairs and two-hop pairs in [21], [24]. Moreover, the sampling techniques in [21], [24] are asymptotically unbiased.

Our contributions in this paper differ from the existing works on several fronts. Firstly, many existing graph sampling algorithms are overly complicated, which contradicts our goal of leveraging graph sampling to accelerate graph mining tasks. For example, Benczur et al. propose to preserve minimum cuts via NURE Sampling in [4], which can get rid of the strong condition in the original method with URE Sampling. However, the NURE Sampling process alone is very time-consuming as it needs to compute the connectivity of each edge. Secondly, in terms of the application of graph sampling to the problem of community detection, our approach is more general than those proposed by [19], [26] and [27] as we adopt URE sampling as a preprocessing step which is applicable to any community detection algorithm. Thirdly, our work extends the scope of Karger's theoretical guarantee [12], [13] and broaden its applications beyond max-flow/min-cut problems to cover additional graph analysis/mining tasks.

III. Graph Properties Preservation via URE Sampling

In this section, we mainly present the edge-set metrics extended from the cut-set and show the theoretical results for graph property preservation under URE sampling.

A. Definitions and Preliminaries

In this paper, we focus on the social network with undirected relationships and model it as a graph $G = (V, E)$, where $V = \{v_1, v_2, \ldots, v_n\}$ is the vertex set and $E = \{e_1, e_2, \ldots, e_m\}$ is the edge-set. To facilitate discussions in the subsequent sections, we first define the following concepts and notations:

Definition 1. *Given an undirected graph $G = (V, E)$, the weight of an arbitrary subset of edges $F \subseteq V \times V$ is defined as*

$$w(F) = \sum_{e \in F} w(e)$$

where $w(e)$ is the weight of a single edge. For those edges $e \notin E$, we let $w(e) = 0$. For an unweighted graph, we let $w(e) = 1, \forall e \in E$.

Definition 2. *Given an undirected graph $G = (V, E)$, we extend the cut-set to yield the following 5 types of edge-set where S is a subset of V and $\bar{S} = V - S$.*

- *Volume:* $v(S) = \{(u, v) \in E | u \in S\} = \{(u, v) | u \in S\} \cap E$
- *Association:* $\rho(S) = \{(u, v) \in E | u \in S, v \in S\} = \{(u, v) | u \in S, v \in S\} \cap E$

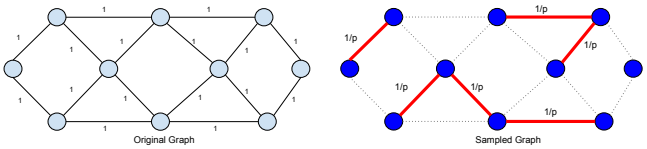


Figure 1: Uniform Random Edge Sampling

- *Cut:* $\delta(S) = \{(u,v) \in E | u \in S, v \in \bar{S}\} = \{(u,v) | u \in S, v \in \bar{S}\} \cap E$
- *Complement Volume:* $\bar{v}(S) = \{(u,v) \in E | u \in \bar{S}\} = \{(u,v) | u \in \bar{S}\} \cap E$
- *Complement Association:* $\bar{\rho}(S) = \{(u,v) \in E | u \in \bar{S}, v \in \bar{S}\} = \{(u,v) | u \in \bar{S}, v \in \bar{S}\} \cap E$

For an undirected graph $G = (V, E)$, denote by $G_s = (V_s, E_s)$ the sampled graph obtained from graph sampling. As an illustrative example, Fig. 1 shows the whole process of URE sampling. It first selects all the nodes (i.e., $V_s = V$) and samples a set of edges E_s from E uniformly at random with certain probability p. Suppose the weight of each edge in the original graph is 1, it then sets the weight of each edge in G_s to $1/p$. The resultant graph is the desired sample of the original graph from URE sampling.

B. Property Preservation under URE Sampling

Applying the same approach from Karger's work, we prove in this section that, for all of the five types of edge-set defined in the previous subsection, their weights (See Definition 1) can be well preserved under URE sampling. To establish this result, we first prove the following theorem, which provides an upper bound for the number of edge-sets with specific constraints. In the following description, when we talk about the metric preservation for these five types of edge-set, we refer to the preservation of their weights.

Theorem 1. *Given an undirected graph $G = (V, E)$, the number of different edge-sets with $|F| \leq \alpha c$ for each type is less than $n^{2\alpha}$, where c is the minimum node-degree in G.*

Combine the result in Theorem 1 and apply the same approach in [12], the following theorem immediately follows:

Theorem 2. *Given an undirected graph $G = (V, E)$, let n be the number of vertices in the graph and $c = \Omega(\ln n)$ be the minimum node-degree. We sample edges of G with probability $p = \frac{3(d+2)\ln n}{\epsilon^2 c}$ (d is a positive number) independently and set their weights in G_s as the original multiplied by $1/p$. Then G_s is an ϵ-approximation of G under all of the five types of edge-set metrics. More precisely, if we denote by F_G and F_{G_s} a particular edge-set in the original graph G and the corresponding sampled graph G_s respectively, then for any F_G with $|F_G| \geq c$, the following inequality holds with a high probability:*

$$(1-\epsilon)w(F_G) \leq w(F_{G_s}) \leq (1+\epsilon)w(F_G) \tag{1}$$

C. Property Extension

Based on the definition of the five types of edge-set and Theorem 2, we can further extend the above preservation results to the following four graph-theoretic properties associated with any subset of nodes, say S, of a graph:

- Ratio Cut [25]: $Rcut(S) = \frac{|\delta(S)|}{|S|}$
- Ratio Association [23]: $Rassoc(S) = \frac{|\rho(S)|}{|S|}$
- Normalized Cut [23]: $Ncut(S) = \frac{|\delta(S)|}{|v(S)|}$
- Normalized Association [23]: $Nassoc(S) = \frac{|\rho(S)|}{|v(S)|}$

Table I: Information of Different Datasets

Name	Nodes	Edges	Minimum Degree
BlogCatalog	10,312	333,983	1
loc-Gowalla	196,591	950,327	1
Flickr	80,513	5,899,882	2
Friendster	5,689,498	14,067,887	1
ego-Facebook	4,039	88,234	

Theorem 3. *Given an undirected graph $G = (V, E)$, let n be the number of vertices in the graph and $c = \Omega(\ln n)$ be the minimum degree of vertices. We sample edges of G with probability $p = \frac{3(d+2)\ln n}{\epsilon^2 c}$ independently and set their weights in G_s as the original multiplied by $1/p$. Then G_s is an ϵ-approximation of G in terms of Ratio Cut, Ratio Association and a 4ϵ-approximation of G in terms of Normalized Cut, Normalized Association.*

IV. Experimental Evaluation

In this section, we present experimental results to evaluate the validity of our theoretical results in Section III when applying URE sampling on large-scale real-world social graphs in which the minimum node-degree requirement is not satisfied.

A. Dataset

We conducted our evaluation using four real world datasets obtained from the Stanford Large Network Dataset Collection[1] and the Social Computing Data Repository of Arizona State University[2]. The statistics of the datasets are shown in Table I.

B. Preservation of edge-set metrics indexed by a randomly generated vertex set

We randomly select a subset of vertices within a given social graph to check whether the metrics of the five types of edge-set (per Definition 2) for this particular vertex subset are well preserved in the URE-sampled graph. We use the Normalized Root Mean Square Error (NRMSE) as the metric to evaluate the extent of graph property preservation. In particular, NRMSE quantifies the relative error of an estimator $\hat{\theta}$ with respect to its true value θ and it is defined as:

$$NRMSE(\hat{\theta}) = \frac{\sqrt{E[(\hat{\theta}-\theta)^2]}}{\theta}. \tag{2}$$

When $\hat{\theta}$ is an unbiased estimator of θ, $E[\hat{\theta}] = \theta$, $NRMSE(\hat{\theta}) = std(\hat{\theta})/\theta$. In each of our experiment, the empirical NRMSE values are obtained by averaging over 100 different runs.

[1] http://snap.stanford.edu/data/

[2] http://socialcomputing.asu.edu/pages/datasets

In Fig. 2a, we show the results of NRMSE at different sampling rate (fraction) for the loc-Gowalla dataset. Observe that the weights of all of the five types of edge-set are well preserved. The NRMSE for these five metrics already approaches a small value of around 0.01 for edge sampling rate as low as 1 percent. The values are all below 0.01 and approach 0.005 if over 10 percent of edges are sampled from the original graph. The difference between the five edge-set metrics in the sampled graph and that of the original graph continuously decreases with the increasing sampling size. Therefore, the weights of the five edge-sets indexed by this particular randomly generated vertex set are well preserved. The experimental results of the other three datasets are very similar, so we have not included them here due to space limit.

C. Preservation of all edge-set metrics under URE sampling

In this subsection, we evaluate the applicability of Theorem 2 in a real-world graph setting when its required condition may not be satisified and show that all the five types of edge-set metrics are still well preserved under URE sampling. Given a graph with n vertices, there are 2^{n-1} edge-sets defined for each type. It is computationally infeasible to enumerate all the possible edge-sets and check their preservation for the massive graphs in our dataset. Instead, we uniformly choose 500 edge-sets at random from the pool of a particular type and check the property preservation results of the 500 chosen edge-sets. Following the proof of Theorem 2, the probability that there exists one among the 500 chosen edge-sets such that the weight cannot be bounded by Inequality (1) is less than $\frac{4}{d}n^{-d} \cdot \frac{1000}{2^n}$. For $n \geq 1000$, this number is close to zero.

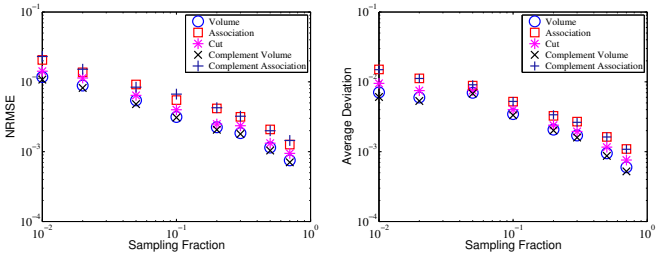
We compare the weight of every edge-set in the sampled graph and that of the original graph to calculate the deviation, namely the value of ϵ in Theorem 2. More precisely,

$$\epsilon = \frac{|w(F_G) - w(F_{G_s})|}{w(F_G)}. \tag{3}$$

We then take the average of the deviations over the 500 edge-set metrics to get the average deviation. Fig. 2b depicts the results of the average deviation at different sampling fraction for the loc-Gowalla graph. Similar results can obtained as in the last subsection. The 500 edge-sets as a whole are already quite well preserved at sampling fraction as low as 1 percent. This is a good indication of the preservation of all edge-set metrics under URE sampling. The experimental results of the other three massive graphs in datasets are very similar as well.

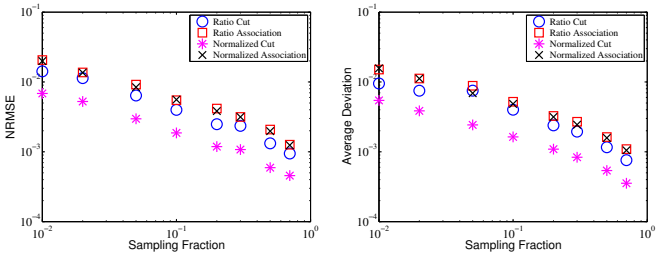
D. Preservation of the four graph-theoretic properties extended from edge-set metrics

In order to demonstrate the validity of Theorem 3 in large-scale graphs in practice, we also present the experimental results for the other four graph-theoretic properties introduced in Section III-C. Fig. 3 depicts the results for the loc-Gowalla graph. Notice that they are also well preserved under URE sampling. The values of NRMSE and average deviation are all around 0.01 at sampling rate as low as 1 percent. Still, similar results can be obtained for the other three social graphs in our dataset.



(a) NRMSE (b) Average Deviation

Figure 2: Preservation results of five edge-set metrics for loc-Gowalla Dataset



(a) NRMSE (b) Average Deviation

Figure 3: Preservation results of four graph-theoretic properties extended from edge-set metrics for loc-Gowalla Dataset

V. Accelerating Graph Mining Algorithms via URE Sampling

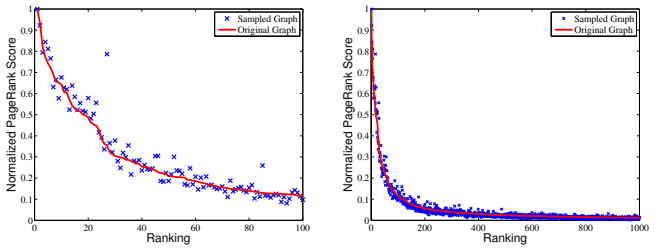
The previous section directly evaluates our extended version of theorems. In this section, we demonstrate how to make use of URE sampling to accelerate common graph mining algorithms, which is the ultimate objective of our work. All algorithms under tested are implemented in Python and executed on a 64-bit Linux server with eight 3.4GHz Intel Core i7 processors and a total of 16 GB RAM.

A. PageRank

PageRank is a popular algorithm used to measure the relative importance of nodes in a graph and the purpose of using PageRank is mainly to get those top ranking nodes. Given an undirected graph $G = (V, E)$ and a vertex v, the PageRank score $r(v)$ of vertex v is defined as:

$$r(v) = (1 - \epsilon) \sum_{(u,v) \in E} \frac{r(u)}{d(u)} + \frac{\epsilon}{|V|}, \tag{4}$$

where $d(u)$ denotes the degree of node u and ϵ denotes the probability of random jump (aka. damping factor). The larger score a vertex has, the higher the vertex ranks and is considered more important. PageRank works by counting the number and quality of links (edges) to a vertex to determine a rough estimate of how important the vertex is. Since URE sampling well preserves all five types of edge-set metrics, the relative importance of vertices should also be well preserved. That is, if a vertex receives many links and connects to more important vertices in the original graph, it should still receive more important links relative to other vertices in the sampled graph.



(a) Top 100 Ranking (b) Top 1000 Ranking

Figure 4: The PageRank scores of the top 100 and 1000 ranking nodes of the original graph and the sampled graph (under the sampling fraction of 20%) for BlogCatalog Dataset

In this subsection, we study the acceleration of the PageRank algorithm through URE sampling. When the original graph is very large, it is time-consuming to execute the PageRank algorithm on the original graph directly. If similar ranking results can be obtained from the sampled graph, the computation time can be substantially shortened. We execute the same PageRank algorithm on the original graph and the sampled graph with a damping factor of 0.15 (the value commonly used in the literature) to compare the results.

The evaluation on the performance of a ranking model is usually carried out by comparing the ranking lists output by the model and the ranking lists given as the ground truth. In our case, the ranking lists obtained by executing the PageRank algorithm on the original graph serve as the ground truth. We illustrate the PageRank scores of top 100 and 1000 ranking nodes from the ground truth of the original graph and the sampled graph (under the sampling rate of 20%) for BlogCatalog Dataset in Fig. 4a and Fig. 4b respectively. It indicates that the PageRank scores of the nodes in the sampled graph do not vary much from those in the original graph.

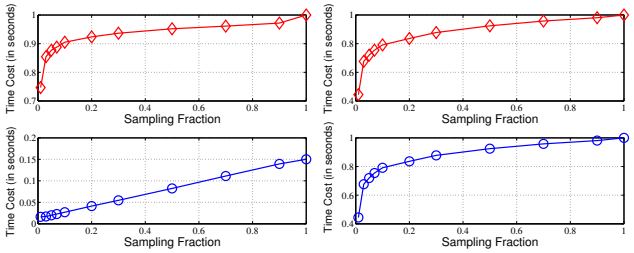
Here, we use Mean Average Precision (MAP) to further quantify the accuracy of URE sampling in computing the PageRank scores. MAP is one of the main measures used in TREC [1] and has been shown to be a stable, effective metric [7]. We consider the top 100 nodes in the original graph as important nodes. In MAP, it is assumed that the grades of importance are at two levels: 1 and 0. And the top-100 ranked nodes have a grade of importance of 1. For a given sampling rate (fraction), we execute the same PageRank algorithm on the sampled graph G_s to produce a ranking list of the top 100 nodes $\pi = \{n_1, n_2, \ldots, n_{100}\}$. The Average Precision for each run is defined as:

$$AP = \frac{\sum_{j=1}^{100} P(j) \cdot y_j}{\sum_{j=1}^{100} y_j}, \tag{5}$$

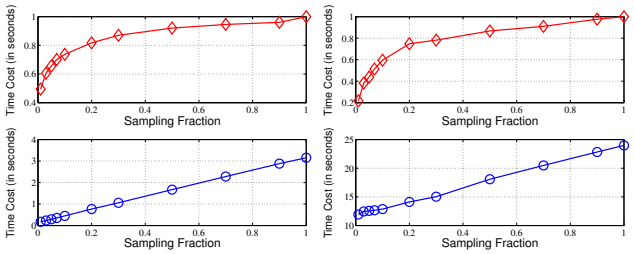
where y_j is the label of node n_j which takes on 1 or 0 as the value, corresponding to the case where a node is important or not. $P(j)$ is defined as:

$$P(j) = \frac{\sum_{k:\pi(k)\leq\pi(j)} y_k}{\pi(j)}, \tag{6}$$

where $\pi(j)$ is the position of node n_j in the ranking list π. Since labels are either of value 1 or 0, 'precision' can be defined. The Average Precision represents averaged precision



(a) BlogCatalog (b) loc-Gowalla



(c) Flickr (d) Friendster

Figure 5: PageRank Algorithm running on four different datasets

over all the positions of nodes with a labelled value of 1. For each sampling rate, the Average Precision values are averaged over 100 runs of the PageRank computation to obtain the Mean Average Precision (MAP).

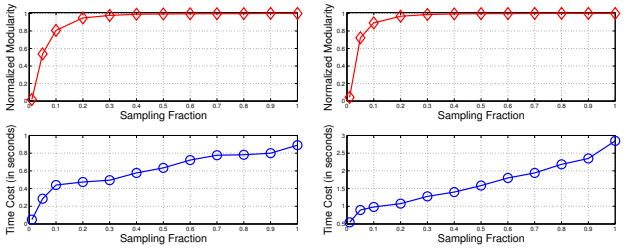
In Fig. 5, we present the experimental results for the four real-world social graphs. Observe that MAP almost reaches 0.9 for the BlogCatalog graph and about 0.8 for loc-Gowalla and Flickr ones respectively when sampling at a rate of as low as 10 percent. For the larger graph of Friendster, MAP also approaches 0.8 at a sampling rate of 20 percent. The computation time for low sampling rate are several times smaller than that for the original graphs. We therefore conclude that, if the purpose of computing PageRank for a graph is to identify and order the top-ranked nodes, it is acceptable to just execute the PageRank algorithm on the URE-sampled graph.

B. Community Detection

Community detection algorithms are heavily used in OSNs to support many of their core services, such as timeline personalization and friend recommendation. These services all require fast discovery of communities. For such applications, speed is much more important than marginal improvement of the quality of community detection results.

On one hand, the preservation of all five types of edge-set metrics will lead to the preservation of densely connected groups. On the other, the relative important group/community structure remains although the number of connections in each group decreases after graph sampling. Therefore, we can naturally predict that we should get similar community detection results by running the community detection algorithms on the sampled graph.

Label Propagation [22] is one of the state-of-the-art community detection algorithms. In this algorithm, every node is initialized with a unique label and at every step each node



(a) LP Algorithm (b) Louvain Method

Figure 6: Performance of URE-based Community Detection using normalized modularity as the metric. (Results are normalized w.r.t. the modularity value (0.8137 and 0.8339 respectively) of the resultant partition of the LP and Louvain algorithms without sampling.)

adopts the label that most of its neighbors currently have. In this iterative process, densely connected groups of nodes form a consensus on a unique label to form communities. Louvain method [6] is another state-of-the-art algorithm for community detection and it has been widely used by industrial practitioners in large scale networks. This method is a greedy optimization approach that attempts to maximize the modularity of a partition of the network.

Modularity is a classical metric to quantify the quality of the results produced by different community detection algorithms under studied [20]. Here we use Modularity to measure the difference in the quality of the results of a community detection algorithm when it is applied to the URE-sampled version of graph instead of the original one. We adopt the definition of Modularity as presented in [2]. In particular, let $\mathcal{C} = \{C_1, C_2, \ldots, C_n\}$ be a partition of the graph $G = (V, E)$ s.t. $C_i \cap C_j = \emptyset$ and $C_1 \cup C_2 \ldots \cup C_n = V$. Then the modularity of the whole graph given $\mathcal{C}$ can be expressed as:

$$Q(\mathcal{C}) = \frac{1}{2m} \sum_{C \in \mathcal{C}} \sum_{u \in C, v \in C} \left(A_{u,v} - \frac{d_u d_v}{2m} \right), \qquad (7)$$

where d_u, d_v denote the degrees of nodes u and v, A is the adjacency matrix of G ($A_{u,v} = 1$ if u and v share an edge and $A_{u,v} = 0$ otherwise). The term $\frac{d_u d_v}{2m}$ computes the expected number of edges between u, v on a Fixed Degree Distribution Random Graph. Thus, for each cluster, it computes the deviation of observed graph from a random graph. If a set of vertices has closer relationships, its modularity should be larger. The summation of modularity values of all sets (communities) is a good indicator of the performance of the corresponding community detection algorithm. The normalization by a factor of 1/2m makes modularity a number from interval [-1,1].

Here, we use the "ego-Facebook" graph [18], which is of good community structure contained, from the Stanford Large Network Dataset Collection to demonstrate the acceleration of the Label Propagation (LP) and Louvain Community Detection algorithms. We execute both of the aforementioned community detection algorithms on the sampled graphs under different sampling rate to obtain the partition (communities) outcome. After that, we use the resultant community-membership of the nodes to partition the original graph and compute the corresponding Modularity values under different algorithms.

Fig. 6 shows the results of performing community detection for the URE-sampled ego-Facebook graph. Note that the modularity value in the y-axes is further normalized w.r.t. the modularity obtained by running the same community detection algorithm over the original graph without sampling (i.e. sampling rate = 1). Observe that, by applying an edge sampling rate as low as 10% for both of the community detection algorithms, the modularity of the resultant partition computed based on the sampled graph is already very close to its counterpart value in the non-sampling case. It is noteworthy that the running time for each sampled case is almost less than 1/2 of that of the non-sampling cases.

VI. Conclusion

This work is a attempt to use Uniform Random Edge (URE) sampling to accelerate graph mining algorithms, which including PageRank Computation and Community Detection. Our primary contribution is to extend Karger's results on preservation of cut-value to other four types of edge-set metrics, and show the theoretical performance bounds as well. We also explain why the preservation of these edge-set metrics are useful for the graph mining algorithms via extensive simulations. Extensions of this work to other graph properties and graph mining algorithms, are likely next steps towards efficient large-scale graph computation.

References

[1] TREC. http://trec.nist.gov/.

[2] G. Agarwal and D. Kempe. Modularity-maximizing graph communities via mathematical programming. *The European Physical Journal B-Condensed Matter and Complex Systems*, 66(3):409–418, 2008.

[3] N. K. Ahmed, J. Neville, and R. R. Kompella. Network sampling designs for relational classification. In *ICWSM*, 2012.

[4] A. A. Benczur and D. R. Karger. Approximating s-t minimum cuts in $\tilde{O}(n^2)$ time. In *STOC*, 1996.

[5] A. A. Benczur and D. R. Karger. Randomized approximation schemes for cuts and flows in capacitated graphs. In *arXiv:cs/0207078*, 2002.

[6] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[7] C. Buckley and E. M. Voorhees. Evaluating evaluation measure stability. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 33–40. ACM, 2000.

[8] D. Chakrabarti, M. Gurevich, and A. Vattani. Preserving pairwise relationships in subgraphs. 2011.

[9] W. S. Fung, R. Hariharan, and N. J. A. Harvey. A general framework for graph sparsification. In *STOC*, 2011.

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

[11] Y. Jia, J. Hoberock, M. Garland, and J. C. Hart. On the visualization of social and other scale-free networks. *Visualization and Computer Graphics, IEEE Transactions on*, 14(6):1285–1292, 2008.

[12] D. R. Karger. Using randomized sparsification to approximate minimum cuts. In *SODA*, volume 94, pages 424–432, 1994.

[13] D. R. Karger. Random sampling in cut, flow, and network design problems. *Mathematics of Operations Research*, 24(2):383–413, 1999.

[14] D. R. Karger and M. S. Levine. Random sampling in residual graphs. In J. H. Reif, editor, *STOC*, pages 63–66. ACM, 2002.

[15] D. R. Karger and C. Stein. An $\tilde{O}(n^2)$ algorithm for minimum cuts. In *STOC*, pages 757–765, 1993.

[16] V. Krishnamurthy, M. Faloutsos, M. Chrobak, J.-H. Cui, L. Lao, and A. G. Percus. Sampling large internet topologies for simulation purposes. *Computer Networks*, 51(15):4284–4302, 2007.

[17] 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.

[18] J. Leskovec and J. J. Mcauley. Learning to discover social circles in ego networks. In *Advances in neural information processing systems*, pages 539–547, 2012.

[19] A. S. Maiya and T. Y. Berger-Wolf. Sampling community structure. In *Proceedings of the 19th international conference on World wide web*, pages 701–710. ACM, 2010.

[20] M. E. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006.

[21] J. C. L. Pinghui Wang, J. Zhao and D. Towsley. Sampling node pairs over large graphs. In *IEEE Int. Conference on Data Engineering (ICDE)*, 2013.

[22] U. N. Raghavan, R. Albert, and S. Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 76(3):036106, 2007.

[23] J. Shi and J. Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, 2000.

[24] P. Wang, J. Zhao, J. C. Lui, D. Towsley, and X. Guan. Unbiased characterization of node pairs over large graphs. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 9(3), 2015.

[25] Y.-C. Wei and C.-K. Cheng. Towards efficient hierarchical designs by ratio cut partitioning. In *IEEE ICCAD*, pages 298–301, 1989.

[26] X. Yu, J. Yang, and Z.-Q. Xie. A semantic overlapping community detection algorithm based on field sampling. *Expert Systems with Applications*, 42(1):366 – 375, 2015.

[27] S. Yun and A.Proutiere. Community detection via random and adaptive sampling. In *COLT*, 2014.

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

APPENDIX

A. Proof of Theorem 1

Proof. Repeat the same contraction process as in [15], the survival probability of each edge-set with $|F| \leq \alpha c$ is at least $n^{-2\alpha}$. Applying the union bound, the result immediately follows. □

B. Proof of Theorem 2

In Theorem 2, we have proved that G_s is an ϵ-approximation of G for the edge-set metrics defined in Definition 2. Based on the extended properties, we have, $Rcut(S) = \frac{|\delta(S)|}{|S|}$ and $Rassoc(S) = \frac{|\rho(S)|}{|S|}$. Multiplying Inequality (1) by $\frac{1}{|S|}$, we can directly show that G_s is also an ϵ-approximation of G in terms of Ratio Cut and Ratio Association. For Normalized Cut and Normalized Association, we prove the results by applying the following lemma:

Lemma 1. *Suppose* $0 < \epsilon < \frac{1}{2}$, *the inequalities* $(1-\epsilon)b \leq a \leq (1+\epsilon)b$ *and* $(1-\epsilon)d \leq c \leq (1+\epsilon)d$ *imply*

$$(1-4\epsilon)\frac{b}{d} \leq (1-2\epsilon)\frac{b}{d} \leq \frac{a}{c} \leq (1+4\epsilon)\frac{b}{d} \tag{8}$$

We first prove that $(1-4\epsilon)\frac{b}{d} \leq (1-2\epsilon)\frac{b}{d} \leq \frac{a}{c}$. By inequalities $(1-\epsilon)b \leq a$ and $c \leq (1+\epsilon)d$, we can obtain

$$\begin{aligned}\frac{a}{c} &\geq \left(\frac{1-\epsilon}{1+\epsilon}\right)\frac{b}{d} = \left(1 - \frac{2\epsilon}{1+\epsilon}\right)\frac{b}{d} \\ &\geq \left(1 - \frac{2\epsilon + 2\epsilon^2}{1+\epsilon}\right)\frac{b}{d} = (1-2\epsilon)\frac{b}{d} \geq (1-4\epsilon)\frac{b}{d}\end{aligned}$$

Then we prove that $\frac{a}{c} \leq (1+4\epsilon)\frac{b}{d}$. Because $0 < \epsilon < \frac{1}{2}$, we can get $2\epsilon(1-2\epsilon) \geq 0$. Then by inequalities $a \leq (1+\epsilon)b$ and $(1-\epsilon)d \leq c$, we can obtain that

$$\begin{aligned}\frac{a}{c} &\leq \left(\frac{1+\epsilon}{1-\epsilon}\right)\frac{b}{d} \leq \left(\frac{1+\epsilon+2\epsilon(1-2\epsilon)}{1-\epsilon}\right)\frac{b}{d} \\ &= \left(\frac{1+3\epsilon-4\epsilon^2}{1-\epsilon}\right)\frac{b}{d} = \left(\frac{(1-\epsilon)(1+4\epsilon)}{1-\epsilon}\right)\frac{b}{d} \\ &= (1+4\epsilon)\frac{b}{d}\end{aligned}$$

This completes the proof. □