




An Overview on Reducing Social Networks' Size

Myriam Jaouadi^(✉)  and Lotfi Ben Romdhane

MARS Research Lab LR17ES05 Higher Institute of Computer Science and Telecom (ISITCom), University of Sousse, Sousse, Tunisia

jaouadimaryem@gmail.com, lotfi.BenRomdhane@isitc.u-sousse.tn

Abstract. Social networks are important dissemination platforms that allow the interchange of ideas. Such networks are omnipresent in our everyday life due to the explosive use of smartphones. Consequently, modern social networks have reached a significant number of users, making their size huge. Thereby scaling over such large data remains a challenging task. Reducing social networks' size is a key task in social network analysis to deal with this data complexity. Many approaches have been developed in this direction. This paper is dedicated to proposing a new taxonomy covering different state-of-the-art methods designed to cope with the explosive growth of social network data. The suggested solution to the extensive generated data is to reduce the network's size. We then categorized existing works into two main classes that reflect how the reduced network is generated. After that, we present new directions for reducing large-scale social network size.

Keywords: Social networks · Graph sampling · Graph coarsening

1 Introduction

Modern social networks have reached an unprecedented number of users [2] due to their accessible handling. For example, Facebook is the first social network to surpass 1 billion registered accounts and currently sits at 2.91 millions monthly active users¹ that tag photos of new friends, check up on old ones, and post about sport, politics, etc [2]. In fact, the classical methods designed for social networks analysis become inapplicable [1]. In order to handle a such problem, several works have been developed with the aim of reducing the network's size while preserving its original properties. Indeed, reducing the network's size will serve to manifold tasks such as influential nodes detection [8], communities selection [1] and so on.

Reducing a social network's size aims at finding a representative pattern of the network while retaining its key properties. Choosing a subset of nodes or/and

¹ <https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/>.

edges from the original network is the simplest way to form the reduced version [3]. Many works have been proposed in this direction. In this paper we provide a state-of-the-art survey on reducing social networks' size. In this detailed survey, we divide the existing models into two main categories, graph sampling and graph coarsening models. Our main concern is to investigate the existing approaches and compare them according to the preservation of the network properties. The recent applications of reducing the network 's size are also surveyed. Finally, future directions are discussed.

The remainder of this paper is organized as follows: Sect. 2 presents preliminary concepts of reducing social networks' size. Section 3 is about graph sampling methods. In Sect. 4 existing models for graph coarsening are discussed. Section 5 outlines recent applications and the final section concludes this paper and proposes some future research directions.

2 Preliminaries

2.1 Problem Definition

Since graphs are the privileged mathematical tools to model social networks, we can define the reduction of a network size as: Given an undirected social graph $G(V, E)$, with $n = |V|$ nodes and $m = |E|$ edges, the goal is to create a version G' having n' nodes such that $n' \ll n$. The reduced graph G' should be the most similar to the initial one G . In other words, G' conserves structural properties of G [2]. We will define graph properties in what follows.

2.2 Network Properties

In order to evaluate the efficiency of the reduction method, we should check some graph properties [5]. Indeed, preserving the original network's structure proves the success of the reduction method. Manifold properties were considered for this task, (e.g. the clustering coefficient, the degree distribution, the graph diameter, etc).

Definition 1 (*Degree Distribution* [2]). *One of the most relevant and simple graph properties is the degree distribution $P_{deg(k)}$ which can be defined as the fraction of nodes in the graph having the same degree k [5]. It can be described formally as:*

$$P_{deg(k)} = \frac{|\{v; deg(v) = k\}|}{n} \quad (1)$$

where $deg(v)$ is the degree of node v .

Definition 2 (*Clustering coefficient* [2]). *Another measure for graph properties is the clustering coefficient which quantifies the likelihood of two neighbors of a node being neighbors themselves. Clustering coefficient $CC(G)$ for a given graph*

G is defined as the ratio of the number of triangles to the number of triplets known as length two paths [6].

$$CC(G) = \frac{3 * \text{numberTriangles}}{0.5 * \text{allTriplets}} \quad (2)$$

where $\text{allTriplets} = \sum_{i=1}^n ((|NB(u_i)| - 1) * |NB(u_i)|)$ with $NB(u_i)$ is the set of direct neighbors of u_i .

Definition 3 (Graph diameter [3]). Diameter $D(G)$ of a graph G can be defined as the longest distance between all pair of vertices in G . More formally, it can be described as;

$$D(G) = \max_{v \in V} R(v) \quad (3)$$

where $R(v)$ is the radius of a node v , i.e. the maximum shortest path distance to all other vertices. Since reducing the network's size will serve to several social network analysis tasks, we can put forth a practical application of such reduction to a well known task which is community detection problem. Multiple approaches have focused on reducing the network's size as a preliminary step for community detection from huge networks. In order to test the effect of the reduction strategy, the clustering quality known as the modularity [7] can be used which is defined as:

$$Q = \frac{1}{2m} \sum_{i,j} (A_{i,j} - \frac{k_i k_j}{2m}) \delta(c_i, c_j) \quad (4)$$

where m is the number of edges in the graph G , A represents its adjacency matrix, k_i denotes the degree of a node i , c_i is the community to which the node i is assigned and the function $\delta(c_i, c_j)$ indicates whether nodes i and j are members of the same community.

A literature review allowed us to distinguish two main families of approaches for reducing the network's size that reflect how the reduced version is generated: graph coarsening and graph sampling. Li-Chun Zang [34] introduced in a recent work a survey on graph sampling as a representation of relevant units of a given graph. The paper talks about sampling from different areas including real graphs, bipartite graphs and conventional graphs. However, there is a lack of a clear categorization of graph sampling approaches. In fact, the author did not illustrate the variety of graph sampling techniques nor their advantages and limits. A recent survey on graph coarsening [33] was proposed with the aim to take a broad look into coarsening techniques. The authors started by showing the several techniques of graph coarsening and its applications in scientific computing with a clear categorization. Then, the emergence of graph coarsening in machine learning, which is of great interest nowadays, was discussed. However, only graph coarsening is presented in the paper for reducing graphs' size, although we distinguish many techniques for reduction. As for our work, the main concern is to investigate the existing approaches and to talk about their advantages and limits. In this detailed survey, we divide the existing models into two main categories, graph sampling and graph coarsening models.

3 Graph Sampling

One of the well known methods for reducing the graph's size is that of graph sampling. The main idea of sampling is to find a representative pattern from the original network while maintaining its properties. Choosing a subset of nodes or edges from the initial graph is the simplest way to create the sample [2, 3]. We distinguish three popular techniques for this family of approaches: Node Sampling, Edge Sampling and Traversal Based Sampling. Figure 1 describes the three sampling techniques for selecting six nodes from the original graph.

3.1 Node Sampling

The aim of node sampling technique is to select a set of k nodes then to retain links between them. The choice of the k nodes can be done randomly. In this direction, Leskovec et al. [5] have developed the well known approach RN (Random Node). It starts by an uniform choice of a subset of nodes and the sample is created on the basis of the selected nodes and edges connecting them. Even its simplicity, RN may create samples with isolated nodes. To overcome this limit, manifold heuristics have been suggested such that RPN (Random PageRank sampling) and RDN (Random Degree Node) [5]. Indeed, the probability of a node being selected is proportional to its PageRank value for RPN and its degree for RDN. Based on degree distribution, Zhu et al. [12] proposed two sampling strategies. The first one called NS-d (Node Degree-Distribution Sampling) tends to create three nodes clusters for high degree nodes, medium degree nodes and low degree nodes using the k-Means algorithm. As for the second strategy, it uses the CountingSort algorithm to sort high degree nodes and then to put them into the reduced graph using a specified sample fraction. Cai et al. [22] proposed two variant algorithms of the existing UNI (uniform sampling) model. In fact, UNI attributes a uniform sampling probability to each node while ignoring the original network's structure. To overcome the inefficiency of UNI and further improve connectivity, the main purpose of the proposed algorithms is to study nodes distribution and connections between them. The first model called AdpUNI divides the userID space into several intervals to make an adaptive change for sampling probabilities. As for the second algorithm AdpUNI+N, it exploits the neighborhood of nodes to obtain a more representative version of the original graph. Based on contextual structures, Zhou et al. [21] have transformed vertices into vectors. Then, in order to have a reduced graph with high quality clusters, nodes are selected from the vectorized space and a sample that maintains graph connectivity is created.

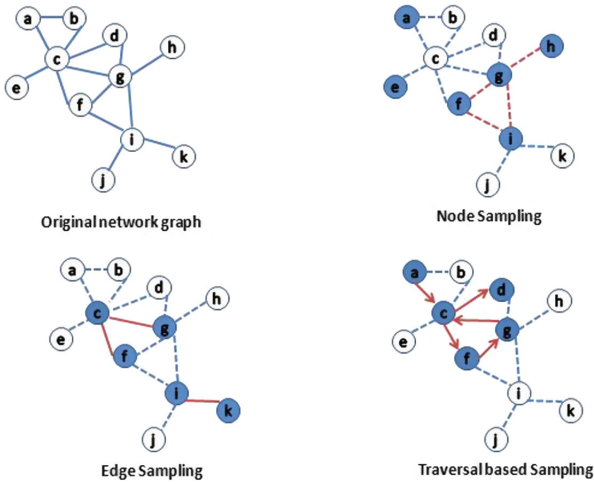


Fig. 1. Graph sampling techniques

3.2 Edge Sampling

Edge sampling is the second technique of sampling. It is based on the idea of choosing a subset of edges at random then including their end nodes in the reduced graph [5]. Random Edge sampling (RE) is the simplest method in this line [3]. During edges selection, only the random chosen ones are added to the sample. A key difference of TIES model (Totally Inducted Edge Sampling) [20] is that, after choosing an initial set of edges and their end nodes, it adds other edges that exist in the original graph among any of this sampled nodes. Wang et al. [9] proposed two algorithms to treat complex networks having a self-similarity structure. In fact, the authors have investigated relations between edges and their neighborhood to choose only those caused by self similarity. DGS (Distributed Graph Sampling) [2] is a recent model designed to sample large scale networks choosing only important edges. Based on the degree centrality, a new measure qualifying the network edges called Edge Importance was proposed. For distribution authors used the MapReduce paradigm [4]. The proposed method demonstrated its efficiency to preserve the original network's structure compared to well known approaches. Similar to DGS, Yanagiya et al. [35] tried to find important edges. The proposed model starts by converting the original graph into a line graph in order to represent edges and connections between them. Then, based on edge smoothness principle, important one are selected to create the sample.

3.3 Traversal Based Sampling

The last family of approaches is traversal based sampling also known as sampling by exploration. Indeed, sampling strategy starts with an initial set of nodes, then

it expands the sample according to some observations. Forest Fire (FF) [5] is inspired by spreading fire in the woods. It begins by picking an initial seed node, then, it burns its outgoing links based on a forward burning probability. The end nodes of the selected edges constitute the next seed set and the process is repeated until reaching the required sample size. RWS (Random Walk Sampling) [15] is one of the most commonly used methods in the literature. It starts by picking an initial seed node at random, then it traverses the graph randomly to move to a random neighbor. Although RW was proved to be simple, in some cases, it gets stuck in an isolated component of the graph. A fast and recent variant called CNARW (Common Neighbor Aware Random Walk) is proposed in order to speed up the convergence of the RW [11]. The basic idea of CNARW is to consider common neighbors of the recent visited nodes to choose next step ones. The proposed scheme reduces RWS cost. DRaWS [16] is another improvement of the random walk model. This work aims to estimate the degree distribution and clique structures while reducing computational costs. In fact, for random walk paths, DRaWS exploits the many-to-one formation between nodes and a clique and the one-to-many formation between one node and many nodes in a clique. DRaWS demonstrates an efficiency in both maintaining the graph's structure and reducing the computational costs. Rank Degree (RD) [13, 15] is a deterministic graph exploration model. Initially, a set of nodes is selected at random. Then, for each node, Rank Degree selects top-k highest degree neighbors and the new nodes are added to the seed set. The process iterates until reaching the desired sample size.

Node Sampling (NS) and Edge Sampling (ES) are very simple. This is because samples are created by selecting a subset of nodes (for NS) or a subset of edges (for ES) randomly or based on some measures such that the degree, the MinCut, etc. However, in some cases we can not apply NS or ES directly due to some constraints like the space [3]. In this case, Traversal Based Sampling (TBS) becomes more suitable for the simple reason that it starts by a few number of seed nodes and then tries to expand it while traversing the network. It is worth noting that the three sampling techniques are not totally different. In fact, some traversal based models are used for NS or ES, for example RWS (Random Walk Sampling) results in uniform edge distribution.

4 Graph Coarsening

Graph coarsening, also known as multilevel approach, is another line of research for reducing the network's size. It is a widely used technique for the resolution of large scale classification problems. The principle aim of coarsening is to convert the original network, level by level, into smaller ones. In other terms, starting from the original graph G , a series of decreasing graphs' size is created. At each level i , the construction of G_i is based on the graph G_{i-1} generated at the previous level [2]. In fact, vertices and edges are collapsed to form the new graph. The choice of vertices or edges to be collapsed can be decided according to several heuristics [1]. RM (Random Matching) [18] is a first implementation of the multilevel approach. It starts by the selection of a non-contracted node randomly.

Then, it matches the selected node with one of its uncontracted neighbors. As a result, the two nodes are merged into one super-node and the algorithm updates the graph's weight. The procedure is stopped when all vertices are marked. In the same direction, Hendrickson and Leland [18] presented an algorithm that starts by constructing a sequence of contracted graphs while adding weights on links and nodes. It is achieved using the maximal cut. Then, the last graph is partitioned by a spectral way. Finally, this grouping is projected and it is periodically improved with the local refinement algorithm Kernighan and Lin (KL) [29]. This algorithm has demonstrated good performance, but it requires a lot of memory to store the graphs during the contraction phase. In fact, another contraction strategy generating fewer intermediate graphs would be a way to reduce this problem. An improvement was introduced in [19] whose particularity lies in the phases of contraction and refinement. Indeed, a new heuristic is presented (hard-edge heuristic) as well as a faster variation of KL is used. As for the partitioning phase, this method tries to find groups of balanced size. As for Edge Matching (EM), it aims to find bisection which minimizes the cutting cost. LEM (Light Edge Matching) [19] tends to minimize edges' weight. In effect, it selects a node u at random. After that, it chooses an uncontracted neighbor v in such a way that the weight $w(u, v)$ of the edge (u, v) is minimal. Chen et al. [27] proposed an hybrid approach based on three matching schemes. First, this approach merges two randomly chosen nodes while keeping the links between them, which guarantees first-order proximity. To guarantee second-order proximity, the second matching is based on the idea of merging nodes having common neighbors. As for the last scheme, it is designed to combine the two previous ones. Another category of contraction approaches is based on optimizing an objective function. In this regard, MCCA (Multi-level Coarsening Compact Areas) [1] is proposed with the aim to minimize the contraction rate. MCCA is designed to merge well connected nodes at every level, then, to update nodes and edges weight until a stopping criterion is met. The algorithm proceeds in a greedy way and at each iteration the number of nodes and edges is reduced. The graph coarsening schema is stopped when the contraction rate reaches a given threshold. Other works have opted to maximize the quality of partition based on the modularity function. In this light was proposed the work of LaSalle et al. [17]. The principal aim of their method is to visit nodes in a random order and to merge each one with a non visited neighbor while maximizing the modularity variation. If the node has an empty neighboring it is shown as contracted. This algorithm prevents big matching in case of power low degree distribution graph. To overcome this shortcoming, the authors developed another variant called M2M which uses a jump secondary during matching. It follows the same principle as the previous method except that if all the neighbors of a node are marked, the latter will be merged with one of its unmarked neighbors. Therefore, M2M improves the old version to generate a small size sample. The contraction has affected various axes in the analysis of social networks, in particular it has been widely used in network partitioning often known as community detection problem [28]. In this context, the Louvain algorithm [30] has been proposed by alternating between

contraction and partitioning. Based on the modularity optimizer, the authors proposed to assign all the vertices to different partitions according to the optimal modularity. Once arrived at a first optimal situation, the process passes to the higher level while processing each partition as a vertex. The process continues until no improvement in modularity is possible. This method solves the problem of the modularity resolution. However, it offers a result that depends on the order of node processing. To overcome this shortcoming, a Louvain variant (with multi-level refinement) is suggested [31]. Another work called SLM (Smart Local Moving) [32] has been developed with the aim of detecting groups in large graphs while maximizing modularity. SLM gives higher modularity values than Louvain, however it requires more calculation time. All the works mentioned above consider only the network structure. However, semantic measures can be used to merge nodes or/end edges for graph coarsening [14]. Among this measures we can talk about homophily [14], which is designed to understand common interests of the network's users.

Discussion

To conclude, graph sampling was widely used. It is one of the first typical thoughts to process massive data. The performance of such strategy is indeed studied in a number of methods to prove that the initial network's structure is preserved [2]. However, graph sampling has some limits. The first one is that it requires a prior knowledge of the global graph which is impracticable in certain cases as in decentralized social networks. Another limit is that, this approach depends on the network law degree distribution. In other terms, nodes degree has an impact in sampling the network and in some cases we can obtain a dense graph or a sample with a distribution different from that of the original one.

In the other hand, graph coarsening was widely used to process large networks but there is no theoretical proof that the initial network's structure is preserved [2]. Another limit is that in many cases, there is a large portion of well connected components which will lead to a poor quality of graph partitioning. Graph coarsening has been proved to be more suitable for the community detection problem [17, 19, 28].

A summary of the different state-of-the-art approaches is highlighted in Table 1. This table presents the advantages, limits and complexity analysis of each approach. We denote by n the number of nodes of a graph G and by m the number of edges.

5 Recent Directions

In order to reduce the network's size, the approaches mentioned above take as input the structure of the entire network. In some cases, due to the distribution of networks, because of their huge sizes or decentralized controls, it is not possible to access to the whole network. In this context, we present distributed approaches as recent directions. Distributed approaches have demonstrated rapidity of implementation and ability to cope with large scale networks. For example, Zang et al. [23] used MapReduce-Spark to implement three distributed algorithms. The

Table 1. Summary on approaches for reducing networks' size

Family	Approach	Complexity	Advantages	Limits
Coarsening	RM [18]	$O(m)$	- Linear - Simple	- Damage the quality of partition
	Hendrickson et Leland [18]		- Good performance in a variety of graphs	- Significant memory consumption
	LEM [19]	$O(m)$	- Simple	- Contracted graph with high degree nodes
	Chen et al. [27]	$O(n+m\log(n))$	- Treat scale free networks - Conserve the proximity of first and second order	- Parameterized algorithm
	MCCA [1]	$O(m+n\log(n))$	- Conserve the original graph properties - Suitable for large scale networks	- Pseudo-linear time complexity
	M2M [17]	$O(m+n)$	- A small size contracted graph	- Unsuitable for power low degree distribution networks
	SLM [32]		- Improve the modularity of the Louvain algorithm	- Significant calculation time compared to Louvain
Sampling	RN [5]	$O(n)$	- Simple strategy for nodes selection	- Unstable due to its random strategy - The low degree distribution is not conserved
	RDN and RPN [5]	$O(m)$	- Improve the random node choice by using a specified heuristic	- RDN: significant number of high degree nodes - RPN: dense graphs
	Zhu et al. [12]	$O(ktn)+O(m+n)$ with t the number of iterations and k the number of K-Means clusters	- Preserve the low degree distribution	- Limited by the sampling rate
	AdpUNI and AdpUNI+ [22]		- Improve the existing UNI (uniform sampling) model in terms of biased sampling	- The sampling size is manually defined
	Zhou et al. [21]		- Use contextual structures to create the sample	- Samples with high number of clusters
	RE [5]	$O(m)$	- Simple, random choice of edges	- The random strategy may affect the network's structure
	TIES [20]	$O(m)$	- Improve the random choice of RE	- Samples with a significant number of links
	Wang et al. [9]		- Low computational complexity compared to other models	- Networks with self-similarity structure
	DGS	$O(m)$	- Scale over huge networks using distributed edge sampling model - Preserve the original network's properties	
	Yanagiya et al. [35]		- Sampling from large scale graphs	- Experimental study limited on some edge selection methods
	FF [5]		- A small size sample - Conserve structural properties	- Polynomial time complexity
	RWS [15]		- Simple, choose nodes based on random walks	- Risk of getting stuck in an isolated component
	DRaws [16]		- Improve the choice of nodes with the simple random walks strategy	- Require an additional time for shortest paths calculation
	CNARW [11]		- Speed up the convergence of random walk sampling	- Degree distribution deviation for the created samples
	RD [13, 15]		- Deterministic graph exploration	- Parameterized algorithm

choice of Spark [25] is argued by the fact that this paradigm puts intermediate data into memory instead of saving it on disk, which makes it much faster. Spark offers also libraries helping to manipulate data in parallel (e.g. GraphX). Indeed, in this work the input graph is partitioned using the node cutting strategy by Graphx and is distributed over a set of machines. Each machine contains one or more partition(s). For the first distributed sampling algorithm, nodes are selected randomly from each partition until the desired sample size is reached. For the second type (distributed edge sampling), a set of links is randomly selected in the reduced sub-graph. For the last one (topology-based distributed sampling), two steps are developed. The first step is to label some nodes and the second stage is designed to sample the graph based on the assigned labels. The advantage of this work is to draw several comparisons between centralized and distributed approaches. In this direction, Gomes et al. [24] proposed distributed versions of some sampling algorithms. In fact, they implemented four distributed versions of RVS models (Random Vertex Sampling), RES (Random Edge Sampling), NS (Neighborhood Sampling) and RWS (Random Walk Sampling). Distribution is performed using the paradigm Spark. Another recent direction is that of using deep learning to formulate the graph sampling problem as a reinforcement learning process. In this direction, Wu et al. [10] have adopted a deep learning strategy to make agents able to select nodes at each time stamp. Then the sample is formed at the end of an episode by the best nodes. Yang et al. [26] have also investigated the deep reinforcement learning to sample networks using directed associative graph. The purpose of this work is to preserve the connection relation of all samples among all episodes. The proposed model demonstrated its efficiency especially for verifying the directed associative graph criteria.

6 Conclusion

In this paper, we focus on the problem of reducing social networks' size. Many current types of research on this problem are developed, and we have discussed their advantages and limits. Our main contribution is to survey the existing models for such problems and present a clear categorization of them. Recent directions for treating large-scale networks are also presented. For the time being, our immediate concern is to test some efficient models on real-world, large-scale social networks and to present an experimental study.

References

1. Rhouma, D., Ben Romdhane, L.: An efficient multilevel scheme for coarsening large scale social networks. *Appl. Intell.* **48**, 3557–3576 (2018)
2. Jaouadi, M., Ben, R.L.: A distributed model for sampling large scale social networks. *Expert Syst. Appl.* **186**, 115773 (2021)
3. Hu, P., Lau, W.C.: A survey and taxonomy of graph sampling. *CoRR* (2013)
4. Liao, Q., Yang, Y.: Incremental algorithm based on wedge sampling for estimating clustering coefficient with MapReduce. In: 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS), pp. 700–703 (2017)

5. Jure, L., Christos, F.: Sampling from large graphs. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 631–636 (2006)
6. Seshadhri, C., Pinar, A., Kolda, T.: Edge sampling for computing clustering coefficients and triangle counts on large graphs. *Stat. Anal. Data Min.* **7**, 294–307 (2014)
7. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. *Phys. Rev. E Stat. Nonlinear Soft Matter Phys.* **69**, 026113 (2004)
8. Wakisaka, Y., Yamashita, K., Tsugawa, S., Ohsaki, H.: On the effectiveness of random node sampling in influence maximization on unknown graph. In: 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC), pp. 613–618 (2020)
9. Wang, W., Fu, X., Lin, X.: Edge-based sampling for complex network with self-similar structure. In: 2021 IEEE Intl Conference on Parallel and Distributed Processing with Applications, Social Computing and Networking, pp. 955–962 (2021)
10. Wu, M., Zhang, Q., Gao, Y., Li, N.: Graph signal sampling with deep Q-learning. In: 2020 International Conference on Computer Information and Big Data Applications (CIBDA), pp. 450–453 (2020)
11. Wang, R., et al.: Common neighbors matter: fast random walk sampling with common neighbor awareness. *IEEE Trans. Knowl. Data Eng.* (2022)
12. Zhu, J., Li, H., Chen, M., Dai, Z., Zhu, M.: Enhancing stratified graph sampling algorithms based on approximate degree distribution. In: Silhavy, R. (ed.) CSOC2018 2018. AISC, vol. 764, pp. 197–207. Springer, Cham (2019). https://doi.org/10.1007/978-3-319-91189-2_20
13. Salamanos, N., Voudigari, E., Yannakoudakis, E.: Deterministic graph exploration for efficient graph sampling. *Soc. Netw. Anal. Min.* **7**, 1–14 (2017)
14. Khanam, K.Z., Srivastava, G., Mago, V.: The homophily principle in social network analysis: a survey. *Multimed. Tools Appl.* (2022)
15. Voudigari, E., Salamanos, N., Papageorgiou, T., Yannakoudakis, E.: Rank degree: an efficient algorithm for graph sampling. In: 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 120–129 (2016)
16. Zhang, L., Jiang, H., Wang, F., Feng, D.: DRaWS: a dual random-walk based sampling method to efficiently estimate distributions of degree and clique size over social networks. *Knowl.-Based Syst.* **198**, 105891 (2020)
17. LaSalle, D., Karypis, G.: Multi-threaded modularity based graph clustering using the multilevel paradigm. *J. Parallel Distrib. Comput.* **76**, 66–80 (2014)
18. Hendrickson, B., Leland, R.: A multi-level algorithm for partitioning graphs. *Supercomputing 1995: Proceedings of the 1995 ACM/IEEE Conference on Supercomputing*, p. 28 (1995)
19. Karypis, G., Kumar, V.: Multilevel k-way partitioning scheme for irregular graphs. *J. Parallel Distrib. Comput.* **48**, 96–129 (1998)
20. Ahmed, N., Neville, J., Kompella, R.: Network sampling via edge-based node selection with graph induction. Department of Computer Science Technical Reports (2011)
21. Zhou, Z., et al.: Context-aware sampling of large networks via graph representation learning. *IEEE Trans. Vis. Comput. Graph.* **27**, 1709–1719 (2021)
22. Cai, G., Lu, G., Guo, J., Ling, C., Li, R.: Fast representative sampling in large-scale online social networks. *IEEE Access* **8**, 77106–77119 (2020)
23. Zhang, F., Zhang, S., Lightsey, C.: Implementation and evaluation of distributed graph sampling methods with spark. *Electron. Imaging* 1–9 (2018)

24. Gomez, K., Täschner, M., Rostami, M.A., Rost, C., Rahm, E.: Graph sampling with distributed in-memory dataflow systems. *CoRR* (2019)
25. Apache Spark. Apache Spark Lightning-Fast Cluster Computing (2015). [Spark.Apache.Org](https://spark.apache.org). Last accessed April 2022
26. Yang, D., Qin, X., Xu, X., Li, C., Wei, G.: Sample-efficient deep reinforcement learning with directed associative graph. *China Commun.* **18**(6), 100–113 (2021)
27. Chen, H., Perozzi, B., Hu, Y., Skiena, S.: HARP: hierarchical representation learning for networks. *CoRR* (2017)
28. Preen, R.J., Smith, J.: Evolutionary n -level hypergraph partitioning with adaptive coarsening. *IEEE Trans. Evol. Comput.* **23**, 962–971 (2019)
29. Kernighan, B.W., Lin, S.: An efficient heuristic procedure for partitioning graphs. *Bell Syst. Tech. J.* **49**, 291–307 (1970)
30. Blondel, V., Guillaume, J., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. *J. Stat. Mech. Theory Exp.* (2008)
31. Noack, A., Rotta, R.: Multi-level algorithms for modularity clustering. In: Vahrenhold, J. (ed.) SEA 2009. LNCS, vol. 5526, pp. 257–268. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02011-7_24
32. Waltman, L., van Eck, N.J.: A smart local moving algorithm for large-scale modularity-based community detection. *Eur. Phys. J. B* **86**, 1–14 (2013)
33. Chen, J., Saad, Y., Zhang, Z.: Graph coarsening: from scientific computing to machine learning. *SeMA J.* **79**, 187–223 (2022)
34. Zhang, L.-C.: Graph sampling: an introduction. *Surv. Stat.* **83**, 27–37 (2021)
35. Yanagiya, K., Yamada, K., Katsuhara, Y., Takatani, T., Tanaka, Y.: Edge sampling of graphs based on edge smoothness. In: ICASSP 2022 -IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5932–5936 (2022)