Information Sciences 306 (2015) 53–69

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

A community-based sampling method using DPL for online social networks

Seok-Ho Yoon [a], Ki-Nam Kim [a], Jiwon Hong [a], Sang-Wook Kim [a,*], Sunju Park [b]

[a] Department of Electronics and Computer Engineering, Hanyang University, Republic of Korea
[b] School of Business, Yonsei University, Republic of Korea

ARTICLE INFO

Article history:
Received 25 November 2013
Received in revised form 30 January 2015
Accepted 8 February 2015
Available online 12 February 2015

Keywords:
Graph sampling
Online social network
Densification power law

ABSTRACT

In this paper, we propose a new graph sampling method for online social networks that achieves the following. First, a sample graph should reflect the ratio between the number of nodes and the number of edges of the original graph. Second, a sample graph should reflect the topology of the original graph. Third, sample graphs should be consistent with each other when they are sampled from the same original graph. The proposed method employs two techniques: hierarchical community extraction and densification power law. The proposed method partitions the original graph into a set of communities to preserve the topology of the original graph. It also uses the densification power law which captures the ratio between the number of nodes and the number of edges in online social networks. In experiments, we use several real-world online social networks, create sample graphs using the existing methods and ours, and analyze the differences between the sample graph by each sampling method and the original graph.

1. Introduction

There have been significant research interests on online social network analysis [2,9,7,20,37,29,35,28,3,36,8,10,27]. As typical online social networks consist of millions of participants, it is almost impossible to analyze them in their entirety [25,22]. What we need is a sample graph representative of the original network [15,22,12,21,33].

Existing graph sampling methods can be classified into three groups: sampling by node selection, sampling by edge selection, and sampling by exploration [22]. When creating a sample from an original graph, sampling by node selection selects a set of nodes and includes edges connecting them. Sampling by edge selection selects a set of edges and includes nodes connected to them. Sampling by exploration creates a sample by selecting a seed node, selecting some neighboring nodes, adding edges connecting the selected nodes, and continuing to include more nodes starting from neighboring nodes.

All the above existing sampling methods, however, fail to generate a sample that retains the properties of an original graph. Firstly, the node–edge ratio of an original graph is not preserved in a sample. Both sampling by node selection and sampling by exploration select nodes repeatedly until reaching a target number of nodes, and sampling by edge selection selects edges repeatedly until reaching a target number of edges. Since the sample size is determined either by the number of nodes or by the number of edges and not by both, the node–edge ratio of an original graph is not preserved in a sample

* Corresponding author.
E-mail addresses: bogely@agape.hanyang.ac.kr (S.-H. Yoon), kinam@agape.hanyang.ac.kr (K.-N. Kim), nowiz@hanyang.ac.kr (J. Hong), wook@hanyang.ac.kr (S.-W. Kim), boxenju@yonsei.ac.kr (S. Park).

http://dx.doi.org/10.1016/j.ins.2015.02.014
0020-0255/

graph. Secondly, a sample generated by the exploration-based method reflects only part of an original graph near the seed node where sampling is taken place, and thus fails to maintain the topology of an original graph. Thirdly, since node or edge selections in existing methods are done randomly, resulting samples tend to be inconsistent with one another and with an original graph.

In this paper, we propose a novel method that generates a sample graph preserving the node–edge ratio and the topology of an original graph. Furthermore, sample graphs created by our method are consistent with one another.

Our sampling method is based on two concepts: hierarchical community extraction and densification power law (DPL). Hierarchical community extraction partitions an original graph into a set of densely-connected subgraphs (i.e., communities). Dendrogram is used to represent the hierarchy between a set of communities [6,30,31]. A sample subgraph, one for each community, is created by selecting nodes within the community with the probability in proportion to the node degree. When the final sample graph is created by merging sample subgraphs, the edges between sampled subgraphs are sampled based on dendrogram. We use the node–edge ratio given by DPL, which represents the ratio between the number of nodes and the number of edges in real-world social networks [25], as a guideline for including edges in a sample.

By combining hierarchical community-based sampling and DPL, we overcome the problems with existing methods. First, DPL makes a sample reflect the node–edge ratio of both local and entire regions of an original graph. Second, hierarchical community extraction makes a sample maintain the topology of an original graph. Third, since our method considers both the node–edge ratio and the topology of an original graph, the properties of sample graphs obtained from the same original graph tend to be consistent with one another and with the original graph.

Through experiments with diverse real-world online social networks, we demonstrate the effectiveness of our sampling method. As performance metrics, we use five well-known social-network properties: degree distribution, singular value distribution, singular vector distribution, average clustering coefficient distribution, and hop distribution. The difference between existing sampling methods and ours is evaluated using K–S *D*-statistics (Kolmogorov–Smirnov *D*-statistics) [22]. The analyses verify that the properties of a sample graph by our method are the most similar to those of the original graph.

The paper consists of the following. Section 2 introduces existing sampling methods and points out the problems with the existing methods. Section 3 describes the proposed method and presents the detailed process. Section 4 compares the performance of the proposed methods with those of the existing methods through experiments. Section 5 summarizes and concludes the paper.

2. Related work

A representative work on social-network sampling methods is [22]. In [22], the authors proposed various sampling methods and demonstrated the effectiveness of each method via experiments on real social networks. Although RN and RE did not use the characteristics of social networks, these two methods were chosen as the baseline methods for performance comparison with RDN, RPN, RW, RJ, and FF, all of which utilized social-network characteristics. In the following, we point out the drawbacks of these methods with respect to social-network sampling.

2.1. Existing sampling methods

Leskovec and Faloutsos classified the sampling methods into three groups: sampling by random node selection, sampling by random edge selection, and sampling by exploration [22]. Fig. 1 shows an example of sampling by random node selection methods. The nodes and edges with solid lines are the ones selected for the sample graph, and those with dashed lines are the ones not selected. The sample graph in Fig. 1 is created by selecting a set of nodes uniformly at random and then by

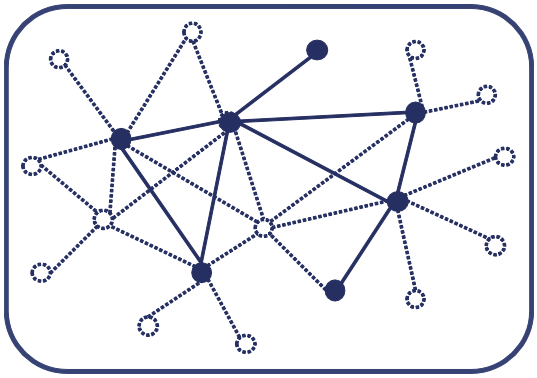

Fig. 1. Sampling methods by random node selection.

selecting all of the edges connecting the selected nodes. If the sample size is given, the method selects nodes repeatedly until satisfying *the number of nodes* to be selected for the sample.

Sampling methods based on random node selection differ in the way nodes are selected. Random Node (RN) sampling selects a set of nodes uniformly at random. The sample graphs created by RN are expected to reflect the properties of the original graph, as samples are selected from the entire population space. The probability of a node being selected is proportional to its degree in Random Degree Node (RDN) sampling, and is proportional to its authority score computed by PageRank [32] in Random PageRank Node (RPN) sampling. The idea behind RDN and RPN is to increase the chance of including *important* nodes in a sample graph. The nodes with many edges, the hubs, are the important nodes in a social network and should be included in a sample graph [4].

Fig. 2 shows an example of sampling by random edge selection methods. The sample graph in the figure is created by selecting a set of edges uniformly at random and then selecting all of the nodes connected to the selected edges. If the sample size is given, the method selects edges until satisfying *the number of edges*.

Similar to sampling by random node selection, sampling by random edge selection differs in the way edges are selected. Random Edge (RE) sampling selects a set of edges uniformly at random and all the nodes connected to the selected edges. The sample graphs created by RE tend not to reflect the structure of the original graph since the high-degree nodes are selected more frequently. Random Node Edge (RNE) sampling solves the problem by selecting a node uniformly at random and then selects some edges uniformly at random among the edges connected to the selected nodes [22].

The sampling methods by exploration create a sample graph by selecting a seed node uniformly at random, exploring its neighbor nodes, selecting all of the nodes explored and their connecting edges, and continuing to explore more nodes. If the sample size is given, the method selects nodes repeatedly from the original graph until satisfying *the number of nodes*.

Depending on which edges to include in the sample, the exploration methods are further classified into two: non-induced and induced. The non-induced method includes only the edges explored in the sample graph. On the other hand, the induced method includes not only explored edges but all edges connected to the selected nodes [12]. Fig. 3 shows the examples of sampling methods by exploration. The node *S* is a *seed node*. The method creates a sample graph by exploring the nodes connected to the seed node, as shown in Fig. 3.

Sampling methods by exploration include Random Walk (RW) sampling, Random Jump (RJ) sampling, and Forest Fire (FF) sampling. Both RW and RJ use the concept of random walk with restart [32]. The difference is the number of seeds used. RW uses a single seed node; RJ uses a set of seed nodes. Compared to FF that explores the graph breadth-first, RW and RJ explore the graph depth-first. FF picks a seed node at random, explores not a single but multiple neighbor nodes. Then, it continues to explore the nodes connected to the explored neighbor nodes recursively.

Table 1 shows the existing sampling methods and their acronyms.

2.2. Problems of existing sampling methods

We point out problems of the existing sampling methods by group. First, sample graphs created by random node selection may have more or fewer *edges* than the number of edges estimated by the ratio between nodes and edges of the original graph, since the sampling methods select nodes until satisfying the estimated number of nodes without regard to the number of edges. For example, suppose that the sample size is 10%, the number of nodes of original graph is 5000, and the number of edges of the graph is 10,000. The sample graph should have 500 nodes and about 800 edges (In social networks the number of edges tends to decrease exponentially when the number of nodes increases linearly. In Section 3, we will explain it in detail). Sampling methods by random node selection do not meet this requirement.

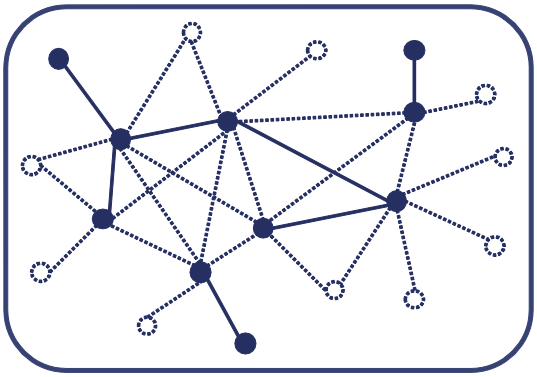

Fig. 2. Sampling methods by random edge selection.

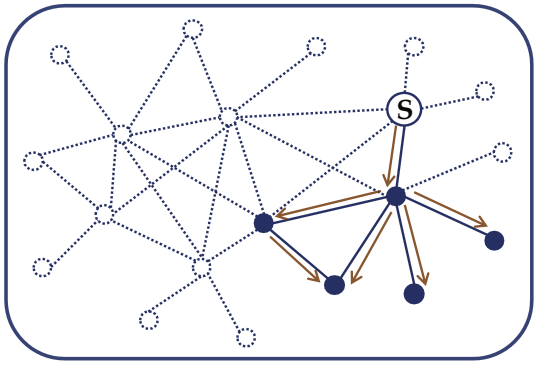

Fig. 3. Sampling methods by exploration.

Table 1
Existing sampling methods.

Acronym	Sampling method
RN	Random Node
RDN	Random Degree Node
RPN	Random PageRank Node
RE	Random Edge
RNE	Random Node Edge
RW	Non-induced Random Walk
RJ	Non-induced Random Jump
FF	Non-induced Forest Fire
RW(i)	Induced Random Walk
RJ(i)	Induced Random Jump
FF(i)	Induced Forest Fire

Most random node selection methods select nodes uniformly at random. The random selection tends to generate *random* samples; the properties of resulting graphs sampled from the same original graph tend to be different with one each another and also with the original graph. RDN and RPN do not select nodes at random but select nodes in proportion to their degrees and authority scores, respectively. Sample graphs created by RDN and RPN are more consistent, but they tend to be denser than the original graph since they include many high-degree nodes.

Second, similar to the cases with random node selection, sample graphs created by random edge selection may have more or fewer *nodes* than the estimation given by the node–edge ratio of the original graph. Also, the properties of the resulting sample graphs created by random edge selection tend to be inconsistent with one another and with the original graph since most methods select edges at random.

Third, sampling methods by exploration repeatedly select nodes until satisfying the estimated number of nodes, as done in random node selection. Thus, similar to the cases with random node selection, sampling by exploration may have more or fewer edges than the estimation provided by the ratio between nodes and edges of the original graph. The node–edge ratio of the sample graph created by the non-induced method is closed to 1:1 since only the explored nodes and edges are selected. The node–edge ratio of the sample graph created by the induced method, on the other hand, is not 1:1 since the explored nodes and all of the edges connecting the selected nodes are selected. All edges connecting the nodes in the graph are included in the sample graph as in the induced methods, which makes the sample graph denser than the original graph. Also, sample graphs created by random edge selection tend to be inconsistent with one another and also with the original graph, since random-edge selection selects edges uniformly at random.

Furthermore, sample graphs created by exploration do not represent entire graph but only the part near the seed node. RJ is regarded as a solution to this problem. Sample graphs created by RJ reflect the properties of the various quarters of the original graph better because it uses multiple seeds.

3. Proposed method

In this section, we propose a new sampling method and describe its process in detail.

3.1. Overview

In the previous section, we have pointed out the problems with existing sampling methods. The new sampling method is designed to achieve the following:

(a) The sample graph should reflect the node–edge ratio of *each region* of the original graph.
(b) The sample graph should reflect the node–edge ratio of the *entire original graph*.
(c) The sample graph should reflect the topology of each region of the original graph.
(d) The sample graph should reflect the topology of the entire graph.
(e) The graphs sampled from the same graph should be consistent with each other.

The properties of a region in a social network may be different from those of other regions and also from those of the entire network. If we create a sample graph that reflects the node–edge ratio of the entire graph only, the node–edge ratio of a particular region in the sample graph may not correctly represent the corresponding region of the original graph. The properties of a graph are closely associated to the topology of the graph. Thus, we should create a sample graph that reflects both the topology of each region and the entire graph. Finally, all sample graphs should be consistent with the original graph so that the properties of each graph are similar to those of the original graph.

To create a sample graph that reflects the properties of the original graph, the proposed sampling method utilizes two key concepts: hierarchical community extraction and densification power law (DPL). First, it uses hierarchical community extraction to partition the original graph into a set of densely-connected subgraphs, i.e., communities. Hierarchical community extraction not only partitions the original graph into a set of communities but also creates a dendrogram that represents the hierarchy of communities. Fig. 4 shows an example of the dendrogram. The large circle is the community, and the edge among the circles represents the parent–child relationship between communities. In Fig. 4, a parent community is partitioned into two children communities. After partitioning the original graph into a set of communities, the proposed method builds sample sub-graphs, one for each community. Then, the method merges sub-graphs into a final sample graph from bottom up, while taking the connections between the communities into account using the dendrogram.

Second, the proposed method uses the DPL, when determining the number of nodes and edges to be included in each sample sub-graph. The DPL states that the number of nodes and the number of edges in a social network follows the power law distribution [25]. Eq. (1) shows DPL, where e denotes the number of edges, and n does the number of nodes. Typically, densification exponent α takes the value between 1 and 2.

$$e \propto n^{\alpha} \tag{1}$$

In the existing sampling methods, the sample size is based on either the number of nodes or the number of edges (but not both). In comparison, the proposed method uses DPL when determining the size of the sample graph. The proposed method estimates the number of sample edges using DPL when the sample size is given as the number of nodes in a social network. The proposed sampling method first determines the value of α for each sub-graph (i.e., community) in the original graph based on the number of edges and nodes within it. It determines the number of nodes to be included in a sample community based on the number of nodes in its corresponding community in the original graph. Then, it computes the number of edges in the sample community by using the value of α for its corresponding original community. A similar approach is used when the sample size is given as the number of edges.

In summary, the proposed sampling method works as follows. First, the method partitions the original graph into sub-graphs. Second, it computes the densification exponent α based on the number of nodes and edges for every community in the original graph. Third, it builds sample sub-graphs by selecting nodes in proportion to their degrees and edges connecting the selected nodes. The node–edge ratio in a community is controlled by α. Finally, it merges sample sub-graphs into a final sample graph in a bottom up fashion, while taking the connections between communities into account. For each merged community, the node–edge ratio is also controlled by α of its corresponding community in the original graph. Fig. 4 shows the process of the proposed method. The nodes and edges in a community with dashed lines are the ones not selected in the sample graph, and those with solid lines are the ones selected in the sample graph. Algorithm 1 shows the pseudo-code.

3.2. Process of the proposed method

In this section, we explain the process of the proposed method in detail.

Algorithm 1. Community-based sampling method using DPL.

```
1 function SAMPLE(G, r_sample)
2     // G: original graph
3     // r_sample: sampling ratio
4
5     // Initialize a sample graph
6     G_s ⇐ ∅
7     // Extract communities from an original graph
8     S ⇐ COMMUNITY_EXTRACTION(G)
9     // Sample from the original communities
10    foreach community s in S
11        // Calculate DPL and the numbers of nodes and edges to be sampled from the community
12        α ⇐ DPL(s.n_nodes, s.n_edges)
13        n_sample_nodes ⇐ s.n_nodes × r_sample
14        n_sample_edges ⇐ n^α_sample_nodes
15        // Sample nodes with a probability proportional to their degree
16        nodes ⇐ SAMPLE_NODES(s, n_sample_nodes)
17        // Sample edges among sampled nodes with a probability proportional to the sum of the degrees of their
              connecting nodes
18        edges ⇐ SAMPLE_EDGES(s, nodes, n_sample_edges)
19        G_s ⇐ G_s ∪ nodes ∪ edges
20    end foreach
21    // Sample edges between sampled communities
22    while S.count > 1 / While there is at least 2 communities in the set of communities
23        // Find the nearest pair of communities and merge them
24        // (Distance measure between two communities is dependent to the community extraction algorithm)
25        P ⇐ NEAREST_PAIR(S)
26        S ⇐ S − P
27        S ⇐ MERGE(P)
28        // Calculate the number of edges to be sampled for connecting two sampled communities
29        P.nodes ⇐ P.nodes ∩ G_s.nodes
30        n_sample_edges ⇐ P.n^α_nodes − P.n_edges
31        // Sample edges among sampled nodes with a probability proportional to the sum of the degrees of their
              connecting nodes
32        edges ⇐ SAMPLE_EDGES(P, P.nodes, n_sample_edges)
33        G_s ⇐ G_s ∪ edges
34    end while
35 end function(G_s)
```

3.2.1. Determining the number of sample nodes and sample edges

The number of sample nodes and the number of sample edges in each community are determined as follows. First, the proposed method computes the number of sample nodes based on the sample size. Second, based on the densification exponent α, the method computes the number of edges to be selected from each community. This process makes sure that the sample graph reflects the node–edge ratio in each community. For example, suppose the sample size is 10%, the number of nodes and edges of in the original graph are 500 and 1000, respectively. Based on Eq. 1, the densification exponent α of the original graph is 1.11. The number of sample nodes and sample edges from that community are 50 and 76, respectively.

All the nodes in the original graph exist within the communities, but some edges in the original graph exist between the communities. Thus, we must determine the number of edges to be selected between two communities (i.e., inter-community edges). The method determines the number of edges to be selected between two child communities as the difference between the number of edges to be selected from the parent community and the number of edges to be selected from two child communities. This provides the number of inter-community edges to be selected between the two communities.

3.2.2. Sampling with communities

The proposed method creates a sample subgraph from each community as follows. First, it selects nodes from each community until satisfying the number of sample nodes predetermined by the sample size. Similar to RDN, the probability of selecting a node is in proportion to its degree, which ensures important nodes are selected. Second, it selects edges until

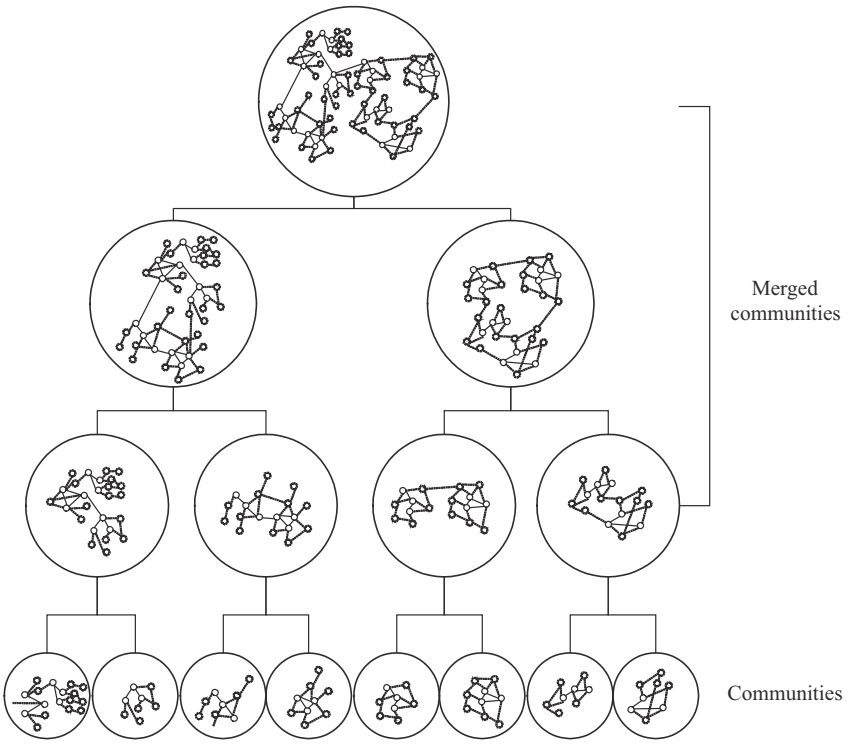

Fig. 4. A dendrogram and the process of the proposed method.

satisfying the number of edges determined by DPL. The probability of selecting an edge is proportional to the sum of the degrees of the nodes connected to it. Sample graphs created by this method are more consistent with one another since the method selects high-degree nodes as in RDN. Also, since the degree of nodes conveys topological information, each sample sub-graph created by this method reflects the topology of each sub-graph of the original graph.

The method creates the final sample graph by sampling edges between communities in the reversed order of partition of the original graph using hierarchical community extraction. There exist a few edges that connect two communities. In social network analysis, these few edges are defined as weak-tie [4]. The method selects the inter-community edges in proportion to the sum of degrees of nodes connected to it, similar to the way it selects edges within the community.

The proposed method may use any hierarchical community extraction method as long as it provides a dendrogram. We use the method which automatically determines the number of communities to be extracted, such as the modularity-based algorithm [6,31] and cross-association (CA) [5]. Of course, one may use the method which requires the number of communities as an input, such as METIS [14] and Chameleon [13], as long as domain experts can supply the optimal number of communities. In this paper, we have used the modularity-based algorithm, a well-known hierarchical community extraction method [6,31].

3.3. Complexity

The proposed method requires additional time for extracting communities from the original graph. The complexity of the modularity-based algorithm is $O(nlogn)$, and the complexity of sampling nodes and edges from the extracted communities is $O(e)$, where n is the number of nodes and e is the number of edges in the original graph. Thus, the total complexity of the proposed method is $max(O(nlogn), O(e))$.

The time complexities of most existing sampling methods are lower than that of the proposed method. The complexities of RDN and RPN, however, are not much lower than that of the proposed method. The complexity of RDN is $O(e)$, since it computes the degrees of all nodes in the original graph. The complexity of RPN is also $O(e)$, since it requires to compute the RageRank scores of all nodes in the original gaph. As $e \propto n^{\alpha}$ in Eq. 1, the complexities of both RDN and RPN are $O(n^{\alpha})$.

A sample graph, once created, can be used over and over in many different analyses. Therefore, when choosing a sampling method, although the time complexity is an important factor, it is more important to have a sampling method that creates a sample graph with the properties similar to those of the original graph. Even if time complexity is higher, the proposed method demonstrates far superior performance in generating a sample graph with properties quite similar to those of an original graph.

Table 2
The size of the dataset used in the experiments.

	# of nodes	# of edges
Wiki-vote	7115	201,524
Email Enron	36,692	367,662
Epinions	75,879	811,480
Hep_ph	34,545	841,754
Hep_th	27,768	704,570
AS	6743	25,144
Oregon	10,669	44,004

4. Experiments

In this section, we demonstrate the effectiveness of the proposed method by comparing it with several existing sampling methods.

4.1. Experimental setup

We use seven real-world online social networks in our experiments [24,23,34,26,19]. First, *Wiki-vote* is a dataset collected from Wikipedia from the day when the service opened to January 2008. A node represents a user of Wikipedia, and an edge represents a recommendation between the users. Second, *Email Enron* is a collection of emails from Enron. A node represents an email address, and an edge represents a communication between email addresses. Third, *Epinions* is a dataset collected from epinions.com, a product review website. A node represents a user, and an edge represents a recommendation between the users. Fourth and fifth, *Hep_ph* (High Energy Physics-Phenomenology) and *Hep_th* (High Energy Physics-Theory) are the datasets from Arxiv website, a website that collected unpublished papers, from January 1992 to April 2003. In both datasets, a node represents a paper, and an edge represents a reference between papers. Sixth, *AS* is a dataset collected from the log analysis of border gateway protocol between the Autonomous Systems, from November 1997 to January 2000. A graph of routers comprising the Internet can be organized into sub-graphs called Autonomous System (AS). A node represents an AS, and an edge represents a communication between ASs. Seventh, *Oregon* is a log data collected from Oregon routers, from March 2001 to May 2001. A node represents a router in Oregon, and an edge represents a communication between the routers. We generate undirected graphs with these data. Table 2 shows the numbers of nodes and edges in each dataset. We use these seven social-network datasets in the experiments and report the average.

Seven performance metrics are used for evaluation. We adopt five major characteristics proposed in [22]: degree distribution, singular value distribution, singular vector distribution, average clustering coefficient (CC) distribution and hop distribution [22]. The degree distribution is the distribution of the number of nodes with degree d for every degree d. The degree distribution of a social network typically follows a power-law distribution [9]. The singular value distribution and the singular vector distribution are the distributions computed by singular value decomposition (SVD) of the graph adjacency matrix [18]. These two properties represent the characteristics of the community structure of a graph. The average CC distribution is the distribution of the average CC of nodes for every degree d. The CC of a node is the ratio between the number of edges among the node and neighbor nodes and the number of possible edges among the node and neighbor nodes. If the number of neighbors of a given node is k, the number of possible edges is $\frac{(k(k-1))}{2}$. The hop is the minimum distance between two nodes. The hop distribution is the distribution of the number of reachable pairs in hop h for every hop h [9]. In addition, we verify how well the *authority* of each node in the original graph is preserved in the sampled graph and how well the community structure in the original graph is preserved in the sampled graph.

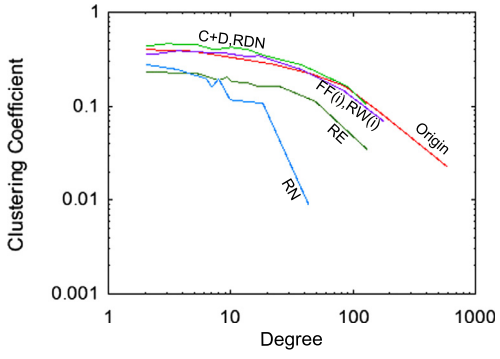

Fig. 5. CC distributions of the original graph and the sample graphs.

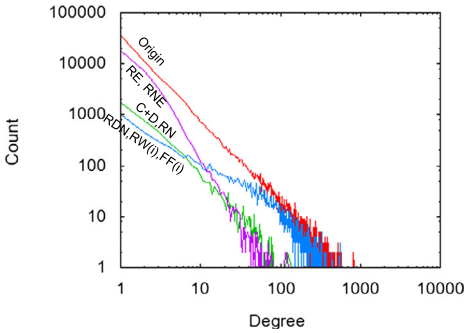

Fig. 6. Degree distributions of the original graphs and the sample graphs.

In experiments, we compare the performance of our propose method with all the methods proposed in [16], RN, RDN, RPN, RE, RNE, RW, RJ, and FF. We use both the non-induced versions of RW, RJ, and FF, and the induced versions of RW (RW(i)), RJ (RJ(i)), and FF (FF(i)), respectively. Our proposed method is denoted as C+D(Mo) (**C**ommunity + **D**PL with **Mo**dularity-based community extraction). To verify that the proposed method works well regardless of the community extraction algorithm, we also added a METIS-based [14] version of the proposed method, C+D(Me). We set the probability of restart to be 0.15 for RW and RJ. For FF, we set P_f to be 0.3, the best value suggested in [22]. In all methods, we use the sample size of 10%.

4.2. Performance comparisons

In this section, we compare the performance of the proposed method with other sampling methods. First, we visually inspect the CC and degree distributions. Second, we examine the performance of various sampling methods more rigorously using K–S *D*-statistics. Third, we measure the similarity between the community structure of the sampled graph generated from each sampling method and that of the original graph. Fourth, we examine whether each node in the original graph keeps its authority in the sampled graph. Fifth, we check the consistency among the graphs sampled from the same original graph by each sampling method. Finally, we examine the densification exponent of each sampling method.

4.2.1. CC and degree distributions

We examine how well the sample graph reflects the properties of the original graph. The evaluation is based on visual inspection on the CC distributions and degree distributions of the sample graphs by various sampling methods. The comparison between the other distributions of the original graph and those of the sampling methods are not presented since they show no visible difference.

Fig. 5 depicts the CC distribution of the original social network and that of each sampled network. In Fig. 5, the *x*-axis represents the degree of nodes, and the *y*-axis represents the CC of nodes. The CC distribution of the original graph is similar to that of the sample graph created by the proposed method. The CC distributions of the sample graphs created by RDN, induced FF, and induced RW are also similar to that of the original graph. The CC distributions of the sample graphs created by RN and RE, however, are quite different from that of the original graph.

Fig. 6 shows the degree distribution of the original social network and that of each sampled network. In Fig. 6, the *x*-axis represents the degree of nodes, and the *y*-axis represents the number of nodes. Again, the shape of the degree distribution of the original graph is similar to that of the sample graph created by the proposed method. The degree distribution of the sample graph created by RN is similar to that of the original graph, but those by the other methods are quite different from that

Table 3
The proposed method vs. node-based sampling methods.

	Degree	*R*	Sval	*R*	Svec	*R*	CC	*R*	Hop	*R*	Avg	*R*
C+D(Mo)	0.132	1	0.044	1	0.176	2	0.338	3	0.045	6	0.147	1
C+D(Me)	0.169	2	0.098	6	0.209	5	0.343	4	0.053	9	0.174	2
RN	0.229	3	0.138	9	0.211	6	0.402	8	0.047	7	0.205	7
RDN	0.258	6	0.085	3	0.215	7	0.367	5	0.032	2	0.191	4
RPN	0.256	5	0.091	5	0.175	1	0.379	6	0.027	1	0.186	3
RW	0.229	3	0.128	8	0.189	4	0.470	9	0.052	8	0.214	8
RJ	0.353	10	0.183	11	0.436	10	0.592	10	0.044	5	0.322	10
FF	0.309	9	0.142	10	0.860	11	1.000	11	0.215	11	0.505	11
RW(i)	0.293	7	0.088	4	0.223	8	0.335	2	0.036	3	0.195	6
RJ(i)	0.506	11	0.104	7	0.185	3	0.380	7	0.036	4	0.242	9
FF(i)	0.305	8	0.084	2	0.265	9	0.260	1	0.058	10	0.195	5

Table 4
The proposed method vs. edge-based sampling methods.

	Degree	*R*	Sval	*R*	Svec	*R*	CC	*R*	Hop	*R*	Avg	*R*
C+D(Mo)	0.128	1	0.041	1	0.187	1	0.303	1	0.042	1	0.140	1
C+D(Me)	0.169	2	0.098	2	0.209	2	0.343	2	0.053	2	0.174	2
RE	0.258	3	0.130	3	0.349	3	0.360	3	0.054	3	0.230	3
RNE	0.302	4	0.143	4	0.580	4	0.518	4	0.061	4	0.320	4

Table 5
The proposed method vs. node-based sampling methods using normalization.

	Degree	*R*	Sval	*R*	Svec	*R*	CC	*R*	Hop	*R*	Avg	*R*
C+D(Mo)	0	1	0	1	0.006	2	0.105	3	0.094	6	0.041	1
C+D(Me)	0.098	2	0.388	6	0.050	5	0.112	4	0.136	9	0.157	2
RN	0.259	3	0.667	9	0.052	6	0.191	8	0.105	7	0.257	7
RDN	0.337	6	0.299	3	0.058	7	0.144	5	0.028	2	0.172	4
RPN	0.332	5	0.340	5	0	1	0.161	6	0	1	0.166	3
RW	0.259	3	0.606	8	0.020	4	0.284	9	0.131	8	0.260	8
RJ	0.591	10	1	11	0.381	10	0.449	10	0.092	5	0.502	10
FF	0.473	9	0.710	10	1	11	1	11	1	11	0.836	11
RW(i)	0.430	7	0.321	4	0.070	8	0.101	2	0.045	3	0.193	5
RJ(i)	1	11	0.433	7	0.015	3	0.162	7	0.046	4	0.331	9
FF(i)	0.463	8	0.292	2	0.132	9	0	1	0.163	10	0.210	6

Table 6
The proposed method vs. edge-based sampling methods using normalization.

	Degree	*R*	Sval	*R*	Svec	*R*	CC	*R*	Hop	*R*	Avg	*R*
C+D(Mo)	0	1	0	1	0	1	0	1	0	1	0	1
C+D(Me)	0.215	2	0.544	2	0.082	2	0.027	2	0.472	2	0.268	2
RE	0.747	3	0.873	3	0.413	3	0.264	3	0.632	3	0.586	3
RNE	1	4	1	4	1	4	1	4	1	4	1	4

of the original graph. The visual inspection suggests that the propose method creates a sample graph with the properties most similar to those of the original graph.

4.2.2. K–S D-statistics

We compute the difference between the five properties of the original graph and those of the sample graph using K–S *D*-statistics (Kolmogorov–Smirnov *D*-statistics). K–S *D*-statistics computes the maximum difference between the cumulative distribution function of the original graph and that of the sample graph (see Eq. (2)). In Eq. (2), x is over the range of the random variable, and F is cumulative distribution functions of the original graph and F' is that of the sample graph. The D value computed by K–S *D*-statistics is between 0 and 1. As the value approaches 0, the property of the sample graph is more similar to that of the original graph.

$$D = max_x(|F(x) - F'(x)|) \tag{2}$$

The size of a sample graph matters; The closer the size of a sample graph is to that of the original graph, the more the properties of the sample graph are similar to those of the original graph [22]. For fair comparison, therefore, we should use sample graphs with the same number of nodes or edges. However, the existing sampling methods determine the size of a sample graph using either the number of nodes or the number of edges. Thus, node-based methods and edge-based methods could not be compared fairly since the number of nodes and edges of the sample graphs created by each sampling method could not be standardized. The proposed method can be compared fairly with both node-based and edge-based sampling methods. Thus, we separate the comparisons into two groups: the comparison between the proposed method and node-based sampling methods and the comparison between the proposed method and edge-based sampling methods. We create 10 sample graphs by each sampling method and then compute the average of the *D*-statistics of the sample graphs.

Tables 3 and 4 show the comparison of the proposed method and node-based sampling methods and the comparison of the proposed method and edge-based sampling methods, respectively. In Tables 3 and 4, the numbers between 0 and 1 represent the *D*-statistics for five different properties, and *R* represents the ranking of the sampling methods. *Avg* represents the average of the five *D*-statistics.

Table 7
Standard deviation of the sample graphs created by various sampling methods.

	Degree	*R*	Sval	*R*	Svec	*R*	CC	*R*	Hop	*R*	Avg	*R*
C+D	0.0002	3	0.0001	7	0.0021	7	0.0005	3	0.0004	6	0.0006	3
RN	0.0053	10	0.0050	11	0.0093	9	0.0230	12	0.0006	10	0.086	10
RDN	0.0002	2	0	3	0.0002	1	0.0003	2	0.0002	1	0.0002	1
RPN	0.0003	4	0	5	0.0002	2	0.0006	4	0.0003	3	0.0003	2
RE	0	1	0	2	0.0003	3	0.0020	6	0.0004	5	0.0006	3
RNE	0.0004	6	0	1	0.0113	10	0.0026	7	0.0004	7	0.0029	8
RW	0.0072	11	0.0007	9	0.0056	8	0.0207	11	0.0005	9	0.0069	9
RJ	0.0004	5	0.0001	6	0.1020	11	0.0109	10	0.0003	4	0.0228	11
FF	0.0092	12	0.0125	12	0.8637	12	0	1	0.0048	12	0.1780	12
RW(i)	0.0024	9	0.0005	8	0.0014	4	0.0070	9	0.0005	8	0.0024	6
RJ(i)	0.0005	7	0	4	0.0018	6	0.0008	5	0.0002	2	0.0007	5
FF(i)	0.0020	8	0.0015	10	0.0016	5	0.0068	8	0.0008	11	0.0025	7

Table 8
Difference between the densification exponent of the sample graph and that of the original graph.

Sampling method	Difference
C+D	0.033
RN	−0.063
RDN	0.135
RPN	0.104
RE	−0.065
RNE	−0.162
RW	−0.096
RJ	−0.142
FF	−0.156
RW(i)	0.141
RJ(i)	0.105
FF(i)	0.149

As shown in Tables 3 and 4, the performance of the proposed method is consistently higher than those of existing sampling methods. In Table 3, the proposed method with modularity-based community extraction, C+D(Mo), is ranked high in all metrics except for the hop distribution. In degree distribution and singular value distribution properties, where the proposed method is ranked the first, the difference between the *D*-statistic values of the top and the second is quite significant. In comparison, the *D*-statistics values of hop distribution, where the proposed method is ranked the sixth, all of the sampling methods are quite similar. Thus, the sample graph created by the proposed method reflects the properties of the original graph the best. In particular, the sample graph created by the proposed method reflects the degree, singular value, and singular vector distributions of the original graph well. This is because the proposed method creates a sample graph by selecting nodes and edges in proportion to their degrees and using the hierarchical community extraction. Note that the *D*-statistics value of CC of non-induced FF is 1. Since non-induced FF selects only the explored nodes and edges and does not explore the explored nodes again, the sample graph created by non-induced FF is a tree, which results in a significant difference in the CC distribution of non-induced FF and that of the original graph.

The proposed method with METIS-based community extraction, C+D(Me), shows results similar to those of C+D(Mo), because both methods take community structure into account when sampling. Compared to C+D(Mo), however, C+D(Me) is ranked relatively low in most performance metrics. We believe this is because modularity-based clustering works better for community extraction than METIS. As shown in Table 4, the proposed method and its METIS version are ranked the first and the second in all metrics.

In Tables 3 and 4, the variance of *D*-statistics in each property differs. When comparing the performance of the sampling methods using the average *D*-statistics of five properties, the overall ranking depends on the *D*-statistics with wide variation. To avoid this problem, we also compare the performance of the sampling methods by normalizing the *D*-statistics. We normalize the *D*-statistics values of each property by min–max normalization [11]. Tables 5 and 6 refer to normalized values and ranks computed with the normalized values. The proposed method outperforms the other sampling methods. The difference between the average score of the top and the second with normalization is larger than that obtained without normalization.

4.2.3. Consistency of sample graphs

In this set of experiments, we evaluate the consistency of various sampling methods. The consistency is measured by the standard deviation of *D*-statistics of sample graphs. As shown in Table 7, the standard deviation of RDN is the lowest, followed by RPN, the proposed method, and RE. Note that the average standard deviations of lower-ranked sampling methods

are quite high. Since RDN and RPN select high-degree nodes, the sample graphs tend to be consistent. RE also selects many high-degree nodes, because it includes the nodes connected to selected edges into the sample graph though edges are selected at random. Thus, the sample graphs created by RE tend to be consistent. The sample graphs created by the proposed method are consistent since it selects nodes and edges in proportion to their degree like RDN and RPN. The sample graphs created by the proposed method is somewhat less consistent than those of RDN and RPN, because RDN and RPN select all of the edges connected to selected nodes while the proposed method selects edges in proportion to degree of nodes connecting the edges. The proposed method generates consistent sample graphs and ranked the third among various sampling methods.

4.2.4. Densification exponent

The sampling target in this paper is a social network. Thus, the ratio of the number of nodes and the number of edges in the sample graph should follow the DPL. In this set of experiments, we examine whether the sample graphs reflect the node–edge ratio of the original graph. Table 8 shows the difference between the densification exponent of the sample graph created by each sampling method and that of the original graph.

The sample graph created by the proposed method reflects the node–edge ratio of the original graph more than the others. The result is not surprising since the proposed method takes into account the node–edge ratio of the original graph when creating a sample graph. Of course, the node–edge ratio of the sample graph created by the proposed method is not exactly the same as that of the original graph. It is because the proposed method uses not the node–edge ratio of the entire graph but the node–edge ratio of each partitioned sub-graph.

The node–edge ratios of the sample graphs created by RN and RE are slightly lower than that of the original graph. In the case of RN, because sample nodes are selected uniformly at random from the entire population space, the selected nodes often do not have edges between them. Similarly, in the case of RE, because sample edges are selected uniformly at random from the entire population, the nodes connected to the selected edges often form an island. In both cases, RN and RE end up with a sample graph sparser than the original graph. RNE is the lowest, because RNE tries to avoid the problem that RE selects many high-degree nodes. In contrast, the node–edge ratio of the sample graphs created by RDN and RPN is much higher than that of the original graph because RDN and RPN select many high-degree nodes.

The sample graphs created by non-induced RW, RJ, and FF are much lower than that of the original graph because the method selects only explored nodes and edges. In contrast, the sample graphs created by induced RW, RJ, and FF are denser than that of the original graph because they select not only explored nodes and edges but also all of edges connecting selected nodes. RJ uses multiple seed nodes while RW and FF use a single seed node. Thus, the node–edge ratio of the sample graph created by RJ is lower than those of RW and RJ even though that of the sample graph created by RJ is denser than that of the original graph.

4.2.5. Community structure

In this set of experiments, we adopt the pair-counting Rand index [1] to measure the degree of similarity between the community structure of the sampled graph and that of the original graph. That is, we examine whether each pair of nodes from the same community in the sampled graph also belongs to the same community in the original graph, and whether each pair of nodes from different communities in the sample graph also belongs to different communities in the original graph. The pair-counting Rand index is defined as follows:

$$RI = \frac{TP + TN}{TP + FP + FN + TN}, \tag{3}$$

where TP (True Positive) is the number of node pairs, each of which has its two nodes in the same cluster in both the sampled graph and the original graph; TN (True Negative) is the number of node pairs, each of which has its two nodes in two different clusters in both the sampled graph and the original graph; FP (False Positive) is the number of node pairs, each of which has its two nodes in different clusters in the original graph but in the same cluster in the sampled graph; and FN (False Negative) is the number of node pairs, each of which has its two nodes in the same cluster in the original graph but in different clusters in the sampled graph.

Table 9 lists the Rand index of various sampling methods. Random selection methods, such as RN and RNE, show better results than the exploration-based methods, such as RW and FF. We conjecture that uniform-random selection of nodes from the entire graph is more advantageous to maintain community structure. Among exploration-based methods, RJ and RJ(i) seem to show better results than the other exploration-based methods due to multiple seeds that are chosen randomly. The proposed method, C+D(Mo), maintains the community structure of the original graph well, since it explicitly considers the community structure when extracting the sample graph. Since METIS performs poorly in community extraction, the Rand index of the METIS version of the proposed method, C+D(Me), is low.

4.2.6. Authoritative nodes

Authority is a metric that estimates the importance [17] of each node in a social network. The authority of a node in a graph can be measured by applying PageRank [32] to the graph. Since the size of the original graph and that of the sampled graph are different, a direct comparison of the authority scores between the nodes from the two graphs would be meaningless. In our experiments, we use Kendall's τ [16] to examine how much the authority rank of each sampled node is preserved in that of the corresponding node in the original graph.

Table 9
Rand index between communities from the sample graph and from the original graph.

Sampling method	*RI*	Rank
C+D(Mo)	0.793	1
C+D(Me)	0.728	10
RN	0.764	4
RDN	0.746	7
RPN	0.761	5
RE	0.741	9
RNE	0.786	2
RW	0.679	12
RJ	0.750	6
FF	0.630	13
RW(i)	0.742	8
RJ(i)	0.773	3
FF(i)	0.701	11

$$\tau = \frac{(number\ of\ concordant\ pairs) - (number\ of\ discordant\ pairs)}{\frac{1}{2}n'(n'-1)} \tag{4}$$

Kendall's τ counts the number of pairwise disagreements between the ranking list of the original graph and that of the sampled graph.

Table 10 shows for each sampling method. RJ(i), RW(i), RDN, and RPN show higher τ values than the proposed method, because these methods tend to sample the nodes with high PageRank scores. RJ(i) an RW(i) add all the nodes and edges that are searched during RWR into the sampled graph, and thus the PageRank order is well maintained in the resulting sampled graph. Although RDN and RPN use PageRank to sample nodes, they add the edges between chosen nodes only, which explains somewhat lower τ values compared to those of RJ(i) and RW(i). The proposed method uses an algorithm similar to RDN when sampling nodes inside a community. The proposed method's τ, however, is lower than RDN, since it samples only a subset of edges.

4.3. The effectiveness of the techniques used in the proposed method

In the previous section, we have shown that the proposed method outperforms the other sampling methods. As the proposed method is based on two techniques, community-based sampling and DPL-based sampling, we examine the effectiveness of these two techniques.

4.3.1. The effectiveness of community-based sampling

First, we examine the effectiveness of the community-based sampling technique when creating a sample graph that reflects the properties of the original graph. We apply the community-based sampling technique to the existing sampling methods and evaluate the performance of the sampling methods with community-based sampling.

For experiments, four representative sampling methods are selected: RN for sampling by random node selection, RE for sampling by random edge selection, RW for sampling by exploration, and RDN whose selection process is similar to that of the proposed method. We call RN, RE, RW, and RDN with community-based sampling as community-based RN,

Table 10
Kendall's τ between PageRank order of the sample graph and that of the original graph.

Sampling method	τ	Rank
C+D(Mo)	0.511	7
C+D(Me)	0.531	6
RN	0.503	8
RDN	0.727	3
RPN	0.703	4
RE	0.434	9
RNE	0.420	10
RW	0.362	11
RJ	0.347	12
FF	0.184	13
RW(i)	0.739	2
RJ(i)	0.745	1
FF(i)	0.697	5

community-based RE, community-based RW, and community-based RDN, respectively. To apply the community-based method to the existing sampling methods, we have done the following. The community-based RN selects nodes at random from each community partitioned by hierarchical community-extraction method and then selects all of edges connecting the selected nodes. The community-based RE selects edges at random from each community and then selects all of nodes connected to the selected edges. The community-based RDN selects nodes in proportion to the degree of a node from each community and then selects all of edges connecting the selected nodes. The community-based RW selects a seed node from each community and then selects all of nodes explored by exploring neighbors of the seed node. It selects all of edges connecting selected nodes. The experimental setup and method are the same as those in Section 4.2.2.

Table 11 compares the performance of the existing methods with and without the application of community-based sampling. In Table 11, *CBased* represents a method with community-based sampling. Table 11 confirms that the community-based sampling methods outperform the original sampling methods.

Community-based sampling makes the sample graph to reflect the topology of the original graph better. For example, the nodes in a community are densely connected to each other. Thus, if community-based RN selects nodes and edges in a community, the nodes in the sample graph can be thought densely connected to each other. Thus, community-based RN creates a sample graph that reflects the properties of the original graph better than RN. The performance of all community-based sampling methods improves for a similar reason. We conclude community-based sampling is an effective technique for creating a sample graph which reflects the properties of the original graph.

4.3.2. The effectiveness of DPL-based sampling

In this section, we examine the effectiveness of the DPL-based sampling technique for creating a sample graph that reflects the properties of the original graph. We apply the DPL-based sampling technique to the existing sampling methods and evaluate the performance of the sampling methods with and without DPL-based sampling.

Similar to the previous experiments, we use RN, RE, RW, and RDN as representative sampling methods. We should make sure the node–edge ratio of the sample graph is the same as that of the original graph. A simple way to achieve this would be to create a sample graph by the original method and then include or remove edges to match the node–edge ratio. Note that the inclusion or removal of nodes would not make the sample graph with the desired node–edge ratio because when nodes are removed, edges connecting the removed nodes are removed automatically. Thus, we insert or remove only *edges* from the sample graph created by the existing method. The number of edges in the sample graph created by RN, RE, and RW are less than the number of edges computed by the node–edge ratio of the original graph. Thus, DPL-based RN, RE, and RW include more edges in proportion to the sum of degrees of two nodes connected to the edges. DPL-based RDN retains the edges in proportion to the sum of degrees of two nodes connected to the edges and removes the rest. The experimental setup and method are equal to those of Section 4.2.2.

Table 12 compares the existing sampling methods with and without DPL-based sampling. In Table 12, *DBased* represents a method with DPL-based sampling. Compared to Table 11, Table 12 shows that not all of the DPL-based sampling methods outperform the original sampling methods. DPL-based RW and DPL-based RDN are better than original RW and RDN, respectively, while original RN and RE are better than DPL-based RN and DPL-based RE, respectively. From these results, one may conclude the DPL-based sampling technique is not effective. This conclusion, however, is incorrect. community-based RDN can be viewed as the proposed method without DPL-based sampling. When we compare the performance of community-based RDN (in Table 11) and that of the proposed method (in Table 3), the proposed method is better than community-based RDN, which indicates the DPL-based sampling technique is effective in the proposed method. A more correct interpretation of the results in Table 12 is that it is difficult to apply DPL-based sampling to those methods. The simple inclusion or removal of edges to the sample graph created by the existing methods fails to keep the key concept of each sampling method.

4.3.3. The performance of the proposed method with different densification exponent

We have not been able to conclude from Table 12 that the DPL-based sampling technique is effective. In this section, we show the effectiveness of the DPL-based sampling technique by comparing the performance of the proposed method with different densification exponent.

Table 11
Comparison of the sampling methods with and without community-based sampling.

	Degree	Sval	Svec	CC	Hop	Avg
RN	0.289	0.138	0.211	0.402	0.047	0.217
CBasedRN	0.273	0.112	0.207	0.372	0.052	0.203
RDN	0.549	0.085	0.215	0.367	0.032	0.250
CBasedRDN	0.537	0.085	0.198	0.384	0.035	0.248
RE	0.210	0.130	0.349	0.360	0.054	0.220
CBasedRE	0.188	0.118	0.339	0.359	0.031	0.207
RW	0.540	0.128	0.189	0.470	0.052	0.269
CBasedRW	0.487	0.090	0.188	0.321	0.037	0.225

Table 12
Comparison of the sampling methods with and without DPL-based sampling.

	Degree	Sval	Svec	CC	Hop	Avg
RN	0.289	0.138	0.211	0.402	0.047	0.217
DBasedRN	0.430	0.160	0.355	0.478	0.215	0.328
RDN	0.549	0.085	0.215	0.367	0.032	0.250
DBasedRDN	0.343	0.312	0.267	0.266	0.028	0.243
RE	0.210	0.130	0.349	0.360	0.054	0.220
DBasedRE	0.395	0.062	0.526	0.717	0.258	0.392
RW	0.540	0.128	0.189	0.470	0.052	0.269
DBasedRW	0.348	0.290	0.259	0.230	0.028	0.231

Table 13
The performance of the proposed method with varying α.

d_α	Degree	Sval	Svec	CC	Hop	Avg	R
−0.5	0.382	0.266	0.431	0.509	0.071	0.319	11
−0.4	0.348	0.191	0.406	0.466	0.059	0.294	10
−0.3	0.279	0.124	0.296	0.401	0.073	0.235	9
−0.2	0.231	0.090	0.296	0.324	0.068	0.202	8
−0.1	0.192	0.058	0.149	0.316	0.043	0.152	2
0	0.132	0.044	0.176	0.338	0.045	0.147	1
0.1	0.149	0.065	0.208	0.379	0.042	0.169	3
0.2	0.167	0.089	0.203	0.402	0.036	0.179	7
0.3	0.160	0.087	0.204	0.393	0.041	0.177	4
0.4	0.159	0.087	0.206	0.396	0.038	0.177	5
0.5	0.163	0.090	0.206	0.398	0.038	0.179	6

In Table 13, d_α represents the difference between the densification exponents of the original graph and the sample graph. Table 13 lists the D-statistics of the proposed method with varying densification exponents and the ranking. When d_α is greater, the sample graph created by the proposed method cannot capture the properties of the original graph. When d_α is smaller, the sample graph reflects the properties of the original graph better. Thus, the closer densification exponent of the sample graph is to that of the original graph, the more the properties of the sample graph are similar to those of the original graph. The results confirm that DPL-based sampling is effective, especially when combined with community-based sampling in the proposed method.

4.4. Execution time

In Section 3.3, we have discussed the complexity of the proposed method and the existing sampling methods. In this section, we measure the actual execution time of sampling methods using real social-network datasets. For this set of experiments, we select our proposed method and one representative sampling method from each group mentioned in Section 2.1. They are RDN for random node selection, RE for random edge selection, and RW for exploration. Table 14 shows the execution time of each sampling method with respect to varying sampling ratios. RDN and RE show very short execution times because of the simplicity in their sampling; they sample a fixed number of nodes randomly. They also maintain constant execution times regardless of the sampling ratio. RW shows an execution time slower than those of RDN and RE, because it is required to randomly explore the graph to sample nodes and edges.

The proposed method performs sampling after the community extraction, and thus shows a longer execution time than the other sampling methods. The execution time without community extraction, C+D w/o CE, confirms that the community extraction takes up most of the execution time in our method. We also verify that the execution time without community

Table 14
Execution time of each sampling method in seconds.

Method	Ratio			
	0.1	0.2	0.3	0.4
C+D	1005.476	1008.237	1010.593	1012.918
C+D w/o CE	10.405	13.166	15.522	17.847
RN	3.682	3.697	3.729	3.916
RDN	4.664	5.07	4.945	5.195
RW	6.567	16.099	41.746	70.496

extraction is linear to the sampling ratio. Note that a sampled graph, once generated, can be used repeatedly in various applications. Thus, we believe it is more important to extract a good sampled graph notwithstanding longer execution time.

5. Conclusions

In this paper, we have proposed a new sampling method that combines two techniques: community-based sampling and DPL-based sampling. First, it partitions the original graph into a set of sub-graphs using hierarchical community extraction. Second, it creates sample sub-graphs. The number of nodes and the number of edges in each sample sub-graph are computed based on DPL. Third, it creates sample sub-graphs by selecting nodes and edges connecting nodes in proportion to their degree within the community. Finally, it builds the final sample graph is created by merging the sample sub-graphs by selecting the edges among the communities.

Through a series of experiments using a set of diverse real-world online social networks, we have demonstrated the effectiveness of the proposed method. The results show that the properties of the sample graph created by our method are the most similar to those of the original graph. We have also demonstrated the effectiveness of the two underlying techniques, community-based sampling and DPL-based sampling, by applying them to existing sampling methods. The results show that community-based sampling improves the performance of the existing sampling methods but DPL-based sampling does not. We have shown, however, the DPL-based sampling technique is effective when combined with community-based sampling.

Acknowledgment

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. NRF-2014R1A2A1A10054151).

References

[1] E. Achtert, S. Goldhofer, H.P. Kriegel, E. Schubert, A. Zimek, Evaluation of clusterings–metrics and visual support, in: Proceedings of the 28th IEEE International Conference on Data Engineering, IEEE, pp. 1285–1288.
[2] R. Albert, H. Jeong, A.L. Barabasi, Internet: diameter of the world-wide web, Nature 401 (1999) 130–131.
[3] D.H. Bae, S.M. Hwang, S.W. Kim, C. Faloutsos, On constructing seminal paper genealogy, IEEE Trans. Cybernet. 44 (2014) 54–65.
[4] A.L. Barabási, J. Frangos, Linked: The New Science Of Networks, 2002.
[5] D. Chakrabarti, S. Papadimitriou, D.S. Modha, C. Faloutsos, Fully automatic cross-associations, in: Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2004, pp. 79–88.
[6] A. Clauset, M.E.J. Newman, C. Moore, Finding community structure in very large networks, Phys. Rev. E 70 (2004) 066111.
[7] S. Das, O. Egecioglu, A. El Abbadi, Anónimos: an lp-based approach for anonymizing weighted social network graphs, IEEE Trans. Knowl. Data Eng. 24 (2012) 590–604.
[8] R. Drezewski, J. Sepielak, W. Filipkowski, The application of social network analysis algorithms in a system supporting money laundering detection, Inform. Sci. 295 (2015) 18–32.
[9] M. Faloutsos, P. Faloutsos, C. Faloutsos, On power-law relationships of the Internet topology, ACM SIGCOMM Comput. Commun. Rev. 29 (1999) 251–262.
[10] J. Ha, S.W. Kim, S.W. Kim, C. Faloutsos, S. Park, An analysis on information diffusion through BlogCast in a blogosphere, Inform. Sci. 290 (2015) 45–62.
[11] J. Han, M. Kamber, J. Pei, Data Mining: Concepts and Techniques, Morgan kaufman, 2006.
[12] C. Hubler, H.P. Kriegel, K. Borgwardt, Z. Ghahramani, Metropolis algorithms for representative subgraph sampling, in: Proceedings of the 8th IEEE International Conference on Data Mining, 2008, pp. 283–292.
[13] G. Karypis, E. Han, V. Kumar, Chameleon: hierarchical clustering using dynamic modeling, Computer 32 (1999) 68–75.
[14] G. Karypis, V. Kumar, A fast and high quality multilevel scheme for partitioning irregular graphs, SIAM J. Sci. Comput. 20 (1998) 359–392.
[15] L. Katzir, E. Liberty, O. Somekh, Estimating sizes of social networks via biased sampling, in: Proceedings of the 20th International Conference on World Wide Web, 2011, pp. 597–606.
[16] M.G. Kendall, A new measure of rank correlation, Biometrika (1938) 81–93.
[17] J.M. Kleinberg, Authoritative sources in a hyperlinked environment, J. ACM 46 (1999) 604–632.
[18] F. Korn, H.V. Jagadish, C. Faloutsos, Efficiently supporting ad hoc queries in large datasets of time sequences, in: Proceedings ACM SIGMOD International Conference on Management of Data, 1997, pp. 289–300.
[19] V. Krishnamurthy, M. Faloutsos, M. Chrobak, J. Cui, L. Lao, A.G. Percus, Sampling large Internet topologies for simulation purposes, Comput. Netw. 51 (2007) 4284–4302.
[20] R. Kumar, J. Novak, A. Tomkins, Structure and evolution of online social networks, in: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2006, pp. 611–617.
[21] S.H. Lee, P. Kim, H. Jeong, Statistical properties of sampled networks, Phys. Rev. E 73 (2006) 016102.
[22] J. Leskovec, C. Faloutsos, Sampling from large graphs, in: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2006, pp. 631–636.
[23] J. Leskovec, D. Huttenlocher, J. Kleinberg, Predicting positive and negative links in online social networks, in: Proceedings of the 19th International Conference on World Wide Web, 2010a, pp. 641–650.
[24] J. Leskovec, D. Huttenlocher, J. Kleinberg, Signed networks in social media, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2010b, pp. 1361–1370.
[25] J. Leskovec, J. Kleinberg, C. Faloutsos, Graphs over time: densification laws, shrinking diameters and possible explanations, in: Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2005, pp. 177–187.
[26] J. Leskovec, M. McGlohon, C. Faloutsos, N. Glance, M. Hurst, Patterns of cascading behavior in large blog graphs, in: Proceedings of the 7th SIAM International Conference on Data Mining, 2007, pp. 551–556.
[27] Y. Li, M. Qian, D. Jin, P. Hui, A.V. Vasilakos, Revealing the efficiency of information diffusion in online social networks of microblog, Inform. Sci. 293 (2015) 383–389.
[28] Y.M. Li, H.W. Hsiao, Y.L. Lee, Recommending social network applications via social filtering mechanisms, Inform. Sci. 239 (2013) 18–30.
[29] S.H. Lim, S.W. Kim, S. Park, J.H. Lee, Determining content power users in a blog network: an approach and its applications, IEEE Trans. Syst. Man Cybernet. Part A: Syst. Hum. 41 (2011) 853–862.
[30] M.E.J. Newman, Analysis of weighted networks, Phys. Rev. E 70 (2004) 056131.

[31] M.E.J. Newman, Fast algorithm for detecting community structure in networks, Phys. Rev. E 69 (2004) 066133.
[32] L. Page, S. Brin, R. Motwani, T. Winograd, The PageRank Citation Ranking: Bringing Order to the Web., Technical Report 1999-66, Stanford InfoLab, 1999.
[33] B. Ribeiro, D. Towsley, Estimating and sampling graphs with multidimensional random walks, in: Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement, 2010, pp. 390–403.
[34] M. Richardson, R. Agrawal, P. Domingos, Trust management for the semantic web, in: Proceedings of the 2nd International Semantic Web Conference, 2003, pp. 351–368.
[35] X. Ying, L. Wu, X. Wu, A spectrum-based framework for quantifying randomness of social networks, IEEE Trans. Knowl. Data Eng. 23 (2011) 1842–1856.
[36] S.H. Yoon, J.S. Kim, J. Ha, S.W. Kim, M. Ryu, H.J. Choi, Link-based similarity measures using reachability vectors, Sci. World J. 2014 (2014).
[37] S.H. Yoon, J.H. Shin, S.W. Kim, S. Park, Extraction of a latent blog community based on subject, in: Proceedings of the 18th ACM International Conference on Information and Knowledge Management, 2009, pp. 1529–1532.