Journal of Visual Languages and Computing 41 (2017) 71–78

Contents lists available at ScienceDirect

Journal of Visual Languages and Computing

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

UNMAT: Visual comparison and exploration of uncertainty in large graph sampling

Tan Tang, Sufei Wang, Yunfeng Li, Bohan Li, Yingcai Wu*

State Key Laboratory of CAD & CG, Zhejiang University, China

ARTICLE INFO

Article history:
Received 18 September 2016
Revised 8 April 2017
Accepted 29 May 2017
Available online 16 June 2017

Keywords:
Uncertainty
Graph sampling
Spread-sheet visualization

ABSTRACT

Graph sampling, simplying the networks while preserving primary graph characteristics, provides a convenient means for exploring large network. During the last few years a variety of graph sampling algorithms have been proposed, and the evaluation and comparison of the algorithms has witnessed a growing interest. Although different tests have been conducted, an important aspect of graph sampling, namely, uncertainty in graph sampling, has been ignored so far. Additionally, existing studies mainly rely on simple statistical analysis and a few relatively small datasets. They may not be applicable to other more complicated graphs with much larger numbers of nodes and edges. Furthermore, while graph clustering is becoming increasingly important, it is still unknown how different sampling algorithms and their associated uncertainty can impact the subsequent graph analysis, such as graph clustering. In this work, we propose an efficient visual analytics framework for measuring the uncertainty from different graph sampling methods and quantifying the influence of the uncertainty in general graph analysis procedures. A spreadsheet-style visualization with rich user interactions is presented to facilitate visual comparison and analysis of multiple graph sampling algorithms. Our framework helps users gain a better understanding of the graph sampling methods in producing uncertainty information. The framework also makes it possible for users to quickly evaluate graph sampling algorithms and select the most appropriate one for their applications.

1. Introduction

Advances in computing and storage technologies in the past few years have made it possible to create, collect, and store very large networks such as online social networks, biological networks, and Internet networks. For example, Facebook, a popular social networking service, has emerged as the world's largest social network, with 800 million users worldwide as of September 2011.[1] While it is relatively straightforward to collect and store the networks, understanding and visualizing these huge networks can be quite challenging because of the increased algorithm complexity, the limited display size, and the constrained human visual processing capability [1].

Random graph sampling has emerged as a useful tool for large graph analysis to circumvent the limitations in recent years [1,2]. It extracts a representative graph from a large network, such that an analyst analyze and explore the much smaller extracted graph rather than the large original graph. The core characteristics of the original network, such as node degree and clustering coefficient distributions, should be preserved in the extracted graph to ensure the effectiveness of the analysis using the extracted graph. Researchers have proposed many different random graph sampling methods such as random walk techniques [3] and breadth-first sampling techniques [4] to preserve the graph characteristics in samples. These methods usually have different strengths and weaknesses in different contexts, which poses a challenge for analysts to select an appropriate graph sampling method for a certain application.

A variety of performance tests [5–7] have been conducted in recent years to evaluate the effectiveness of different graph sampling approaches. Nevertheless, an important aspect of graph sampling, namely, uncertainty from graph sampling due to the variability and lack-of-knowledge, has not been evaluated so far. It is known that random sampling strategies and their possible biases could lead to uncertain and biased samples [8]. Some samples, for example, may maintain the core graph characteristics very well, while other samples produced by the same method may exhibit quite different graph characteristics. If an analyst happens to analyze and explore the graph based on the problematic samples, she is likely to be misled and draw unreliable, even erroneous conclusions.

* Corresponding author. .
E-mail address: ycwu@zju.edu.cn (Y. Wu).

[1] http://en.wikipedia.org/wiki/Facebook

http://dx.doi.org/10.1016/j.jvlc.2017.05.006
1045-926X/© 2017 Elsevier Ltd. All rights reserved.

Existing studies largely rely on simple statistical methods such as D-statistic [5] to measure the effectiveness of the sampling methods in preserving the important features of the original network. Insightful and comprehensive analysis is still absent. While graph clustering has become increasingly important [9], little work has been done for studying how different graph sampling methods can affect the subsequent analysis.

In this work, we propose an efficient visual analytics framework for evaluating different graph sampling methods in analysis of large networks from the aspect of uncertainty. In particular, it assesses the uncertainty for different sampling methods and characterizes the influence of the uncertainty in graph analysis procedures, such as analyzing the node degree or clustering coefficient distributions. We also present a spreadsheet-style visualization to not only quantitatively but also qualitatively compare sampling methods from different viewpoints, such that we can gain a better understanding of the graph sampling methods in producing uncertainty information and influencing the analysis results. With the intuitive visualization and a suit of useful user interactions, the framework allows for sophisticated analysis and comparison of different graph sampling algorithms. Hence, the framework not only allows analysts to choose an appropriate graph sampling approach with less uncertainty, but also enables them to make informed decisions and evaluate the insight gained in the subsequent graph analysis procedures based on the quantified uncertainty. Our contributions are as follows:

- We study a new problem of characterizing impact of uncertainty from sampling on graph analysis
- We propose a general framework for quantifying the uncertainty information derived from graph sampling algorithms.
- We develop spreadsheet-style visualization for exploring and analyzing the uncertainty of extreme-scale graph analysis.

2. Related work

2.1. Graph sampling

Graph sampling is essential for large graph exploration and visualization [1,5]. There are many well-known sampling algorithms such as random node sampling and edge sampling methods [5], *breadth-first sampling (BFS)* [2,4], and *random-walks (RW)* [3] to derive representative samples from large graphs. More sophisticated algorithms such as *Metropolis-Hasting RW (MHRW)* [10] and *Frontier Sampling (FS)* [11] have been also developed to help users obtain better sampled graphs. Nevertheless, most of these sampling methods have their own strengths and weaknesses and it becomes quite difficult for users to choose an appropriate sampling method. Thus, there is a growing demand to evaluate and compare these sampling approaches.

Previous evaluation studies [5,6] often assessed the quality of a sampling method based on how well the graph characteristics (node degree distribution and clustering coefficient distribution) of the original graphs can be preserved in the sampled graphs. Leskovec and Faloutsos [5] systematically compared 10 sampling algorithms including general sampling methods, such as BFS and RW methods, and their new methods, such as *Forest Fire (FF)* and *PageRank (PR)*, using statistical evaluation methods. Their results reveal that the RW methods can obtain the best samples for general static graphs as the methods are biased towards high-degree nodes. As for dynamic graphs, FF and PR methods perform best since they are not as biased and the sampled graphs can preserve the temporal evolution of the original graph. Wang et al. [6] presented a comprehensive comparison among several algorithms including BFS, MHRW, FS, and *Unbiased Sampling in Directed Social Graph (USDSG)* [12]. They quantitively measure the difference between the sampled graphs and the original ones by computing the normalized mean square errors. They found that the performance of different methods highly depends on the graphs used.

Evaluating the sampling algorithms using these methods [5,6] would become very difficult for extreme-scale graphs. These methods need to estimate the graph characteristics of the original graph, which are usually computationally expensive. Recently, Gjoka et al. [10] used uniformly sampled graphs as "ground true" to test several widely-used BFS and RW methods with an extreme-scale social network in terms of convergence and estimation bias. They pointed out that no algorithms is perfect at preserving both node degree and clustering coefficient distributions in different datasets. In contrast to existing evaluation studies, our work tests the performance of different sampling algorithms from the perspective of uncertainty. Furthermore, our study does not require the time-consuming estimation of the original graph features for uncertainty measurement.

2.2. Uncertainty quantification and visualization

Uncertainty Quantification (UQ) is to characterize the impact of data or parameter uncertainty on the outcome of complex systems, such that reliable predictions can be provided [13]. UQ methods can be typically classified into nonsampling methods and sampling methods.

General non-sampling methods such as perturbation methods [14,15], moment analysis [16], and operator-based methods [17] are often employed for uncertainty evaluation. While these methods have been successfully applied to various problems, their applicability is restricted to systems with small uncertainty [18]. In this work, we introduce a variance score method (VSM) to quantify uncertainty of sampling approaches.

Uncertainty can arise from any stage in a data visualization pipeline including data acquisition, transformation, and visualization [19,20]. Showing uncertainty effectively in data visualization can help analysts correctly interpret the data and draw reliable conclusions [21,22]. Pang et al. [19] presented a comprehensive survey of techniques including adding glyphs, adding geometry, modifying geometry, modifying attributes, animation, and psycho-visual approaches to visually represent data together with uncertainty. Thomson et al. [23] suggested a typology for visualizing different types of uncertainty in intelligence analysis. The typology was extended by Zuk and Carpendale [24] to include uncertainty of reasoning. Recently, Correa et al. [15] developed a framework to quantify and propagate uncertainty information in a visual analytics process. Wu et al. [25] extended this framework to demonstrate uncertainty information produced along the process of visual analytics on multidimensional datasets.

Various new uncertainty visualization methods such as ambiguation [21], density plots [26], and summary plots [27] have been developed. Olston and Mackinlay [21] presented a new technique called ambiguation to visualize bounded uncertainty information in charts and graphs. Feng et al. [26] introduced a blurring method based on density plots to visualize uncertain data in scatterplots and parallel coordinates. Potter et al. [27] proposed a summary plot for visualizing the data with uncertainty information using a set of descriptive statistics. To evaluate the effectiveness of different uncertainty visualization techniques, a formal user study [22,28] or theoretical analysis [29] can be used. Sanyal et al. [28] conducted a user to compare four widely used uncertainty visualization methods and found that their performance was highly dependent on tasks performed. Three general perceptual and cognitive theories from Bertin, Tufte, and Ware were applied to analyzing different uncertainty visualizations [29]. Deitrick and Edsall evaluated the influence of uncertainty visualization on decision making by an empirical evaluation, which indicates that un-

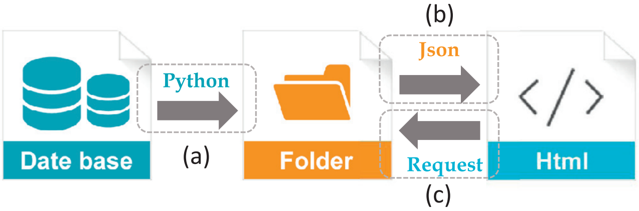

Fig. 1. Visual Analytics Framework containing three stages. (a) Graph processing with Python. (b) Computation outputs stored as Json files. (c) Visualization via web paltform.

certainty visualization can greatly improve the process of decision making [22].

Compared with previous work that focus on developing a new visual representation of uncertainty information, our work mainly aims at characterizing the impact of uncertainty arising from different graph sampling methods and developing a visual analysis platform for visually comparing various sampling techniques from the aspect of uncertainty.

3. Framework

We develop a visual analytics framework to support comparison and exploration of graph sampling uncertainty which is capable of dealing with large-scale graphs. The framework contains three stages. In Fig. 1(a), we select five datasets, namely, amazon, dblp, email, p2p08 as well as p2p31 from Stanford Large Network Dataset Collection [30] and store them in local database for further processing. They help users to compare sampling approaches with various contexts. In Fig. 1(b), We utilize a python library named snap.py [31] to compute graph characteristics (described in Section 4.2). In Fig. 1(c), a visualization platform is developed to explore and understand uncertainty information of various graph sampling approaches. The platform make a request for json objects and get response from a local HTTP server.

4. Graph sampling and analysis

This section describes several commonly used graph sampling algorithms and two graph analysis approaches, which are tested in this work. Assessing the uncertainty from different sampling algorithms in graph analysis may suffer from excessive computational overhead, especially when the graph scales up greatly. Thus, we propose to use parallel computing techniques to accelerate the performance of graph sampling algorithms as well as methods for computing graph characteristics. This enables us to develop an interactive visualization framework for users to analyzing the uncertainty from large graph sampling.

4.1. Graph sampling algorithms

Graph sampling has become increasingly important for large network analysis and exploration. It extracts a representative smaller graph from an original large graph, such that we can use the simplified graph rather than the original graph for further analysis. The goal of graph sampling is to obtain a graph sample as small as possible while preserving the desired graph characteristics. We introduce several popular graph sampling algorithms, including random node sampling, random edge sampling, random waling, and forest fire, to be compared in this work. As our work mainly focuses on very large network data, we use parallel computing techniques to implement the graph sampling algorithms.

4.1.1. Random node sampling

Random node sampling is a straightforward sampling method. It uniformly chooses a set of nodes from a given graph at random. A graph induced by the chosen nodes is the sample of the original graph that we want. Although the method has been criticized for not being able to preserve power-law node degree distribution [5], it has still been widely used and sometimes is employed as a “Ground True” for testing other sampling methods when the original is not available [10].

4.1.2. Random edge sampling

Random edge sampling uniformly selects a set of edges at random and a sample is then a graph induced by the edges, which is similar to random node sampling. While this algorithm is simple, the graph samples could be very sparsely connected and will not retain the community structure shown in the original graph.

4.1.3. Random walking and forest fire

The random walking algorithm and its variants are often used for crawling huge online social networks [10]. The basic idea of random walking is quite simple. It starts by uniformly at random selecting an initial node and then begins a random walk on the graph from the initial node. At each step, there is a pre-specified probability $p = 0.15$ following most other studies in the literature that the algorithm directly goes back to the initial node and restarts the random walk. Furthermore, the algorithm can pick another initial node if the number of obtained nodes is still far less than the required sample size after many steps.

The forest fire sampling is a variant of the random walking algorithm for preserving the heavy-tailed in-degree and out-degree distributions. The algorithm has two parameters, a forward burning probability p and a backward burning ratio r. It first randomly chooses a starting node v and then recursively simulates a “burning effect” for burning the links and the associated nodes. The algorithm detail is listed as follows:

1. Choose a seed node v at random.
2. Generate two random numbers, x and y, geometrically distributed with means $\frac{p}{1-p}$ and $\frac{rp}{1-rp}$, respectively.
3. Node v selects x outgoing links and y incoming links incident to nodes that were not yet visited. Let $\{w_1, w_2, \cdots, w_{x+y}\}$ denotes the other ends of these selected links. Node v selects as many links as possible if there is insufficient outgoing or incoming links.
4. Apply (2) and (3) recursively to each of $\{w_1, w_2, \cdots, w_{x+y}\}$ until sufficient nodes have been burned.

If the fires die before the required graph sample is obtained, the algorithm restarts by selecting a new seed node.

4.2. Graph characteristics

This section describes two commonly accepted graph characteristics (node degree distribution and clustering coefficient distribution), which play important roles in analyzing and exploring networks. They are often used as standard criteria for measuring the effectiveness of graph sampling algorithms in preserving the graph characteristics. Our goal is to determine the impacts of the uncertainty from sampling on the derived graph properties. We also present the efficient parallel methods for deriving the graph characteristics from a large graph.

4.2.1. Node degree distribution

Node degree distribution is one of the most important graph characteristics. The degree of a node in a graph is defined as the number of edges incident to the node. The degree distribution P is defined as follows: the value of the distribution at $k \in \{0, 1, 2,$

$\cdots\}$, namely, $P(k)$, represents the ratio of the nodes with degree k to total nodes in the graph. That is $p(k) = n_k/n$, where n is the total number of nodes and n_k indicates the number of nodes with degree k in the graph. Given a directed graph, its nodes have two different types of degree, in-degree and out-degree, representing the number of incoming edges and outgoing edges, respectively. Thus, there are two types of node degree distribution, in-degree distribution and out-degree distribution, corresponding to node in-degree and node out-degree, respectively. The node degree distribution provides a very important metric for graph analysis, especially for finding a scale-free network, which is defined as a network with its node degree distribution following a power law.

4.2.2. Clustering coefficient distribution

Clustering coefficient distribution is another commonly used graph characteristics. Local clustering coefficient of a node in a graph measures how close its neighbors are to being a complete graph. Given a node v of a directed graph, its local clustering coefficient c_v can be computed as

$$c_v = \begin{cases} \frac{|\{e_{ij}\}|}{d_v(d_v-1)} : v_i, v_j \in N_v & \text{if } d_v > 1 \\ 0 & \text{otherwise} \end{cases}$$

where N_v represents the neighbors of v, $|\{e_{ij}\}|$ is the number of edges among the neighbors of v, and d_v represents the degree of v.

In an undirected graph, we have $e_{ij} = e_{ji}$. There could exist $\frac{d_v(d_v-1)}{2}$ edges among the nodes within the neighborhood of node v. Edges could exist among the vertices within the neighborhood. Thus, the local clustering coefficient c_v for node v of a directed graph can be computed as

$$c_v = \begin{cases} \frac{2|\{e_{ij}\}|}{d_v(d_v-1)} : v_i, v_j \in N_v & \text{if } d_v > 1 \\ 0 & \text{otherwise} \end{cases}$$

The cumulative clustering coefficient distribution P can be defined as: the value of the distribution at $c \in \{0, 0.1, 0.2, \cdots 1.0\}$, namely, $P(c)$, represents the ratio of the nodes with local clustering coefficient less or equal to c to total nodes in the graph. That is $p(c) = n_c/n$, where n is the total number of nodes and n_c indicates the number of nodes with local clustering coefficient less or equal to c in the graph. The clustering coefficient provides a useful tool to determine whether a graph is a small-world network.

5. Uncertainty quantification and visualization

Uncertainty quantification (UQ) aims at characterizing the impact of uncertainty (due to variability and/or lack-of-knowledge) of some system aspects (parameters and/or inputs) on the system outcomes. This section introduces the basic knowledge of uncertainty and its classification, and then presents our visual design called stair-shape chart and UQ method for quantitatively evaluating the uncertainty from graph sampling. We also develop a spreadsheet visualization for systematic comparison of different graph sampling approaches, which allows for users to gain insight of the obtained uncertainty information.

5.1. Stair-shape chart and UQ

Instead of utilizing simple statistics such as D-statistic to describe uncertainty information, we develop a visual design called stair-shape chart to provide deep insights from various sampling approaches intuitively. Two graph characteristic distributions, namely node degree and clustering coefficient, are employed to quantify uncertainty arising from sampling large graphs. Considering that both of them are discrete probability distributions, we adopt a stair-shape line to visualize node degree distribution and clustering coefficient distribution which is a conventional method in mathematical sciences. Fig. 2(a) visualizing node degree distribution whose domain is a integer collections (eg. 1–5). Similarly, clustering coefficient distribution can be visualized with a stair-shape line by slightly changing its domain.

We regard the variance of graph samples' characteristics like node degree distribution as the uncertainty of sampling algorithms. For each pair of sampling approach and dataset, we control sampling rate at twenty percent and implement the sampling algorithm iteratively to generate enough samples for further study. After rendering all stair-shape lines from sample collection, we get a samples chart which reflects the variance of graph characteristic. In Fig. 2(b), every stair-shape line represents certain graph sample and all of these samples lines locate in different places. If these stair-shape lines are compact, we can conclude that their corresponding graph samples are consistent and the proposed sampling algorithm contains fewer uncertainty. In other words, the more centralized a stair-shape chart looks like, the fewer uncertainty a corresponding sampling algorithm has.

Once a samples chart generated, two extreme lines called top-line and base-line are detected automatically and colored by red and blue respectively. Then, all lines in a sample chart subtract their base-line to generate an evolved samples chart, as shown in Fig. 2(c). Because what interests people is relative changes that can represent uncertainty information rather than absolute values, we deliberately ignore coordinates and use round rectangle to enclose all stair-shape lines, as shown in Fig. 2(d). Inspired by the process of handling graph samples data, we name our design as stair-shape chart consisting of plenty of stair-shape lines.

```
initialize a coordinates $X$;
specify the graph characteristic $F$;
set desired graph samples numbers $N$;
while run times smaller than $N$ do
    Generate a graph sample $g$;
    Compute a characteristic distribution (node degree $F_d$ or
    cluster coefficient $F_c$) with $g$;
    if $F$ equals to $F_d$ then
        Add a stair-shape line of $F_d$ in $X$;
    else
        Add a stair-shape line of $F_c$ in $X$;
    end
end
```

Algorithm 1: Stair-shape chart generation.

We implement an effective approach (Algorithm 1) to produce stair-shape chart for a large number of graph samples. Our visual analytics framework calculates uncertainty quantification (UQ) based on stair-shape chart. As mentioned before, two extreme lines, namely, top-line and base-line are generated automatically after the stair-shape chart is produced successfully. The uncertainty quantification employs a variance score method (VSM) to assign each stair-shape chart a value. The more larger VSM value is, the more uncertainty a stair-shape chart has.

$$VSMValue_{Chart} = \sum (h_i^{top} - h_i^{base}) \tag{1}$$

where h_i^{top} represents the height of ith small horizontal line in top-line and h_i^{base} represents the height of ith small horizontal line in base-line.

5.2. Spreadsheet-style visualization

We develop a spreadsheet-style visualization platform called UNMAT to help analysts compare the graph sampling approaches from different uncertainty perspectives. Spreadsheet-style visualizations are well known for their capability of visual comparison

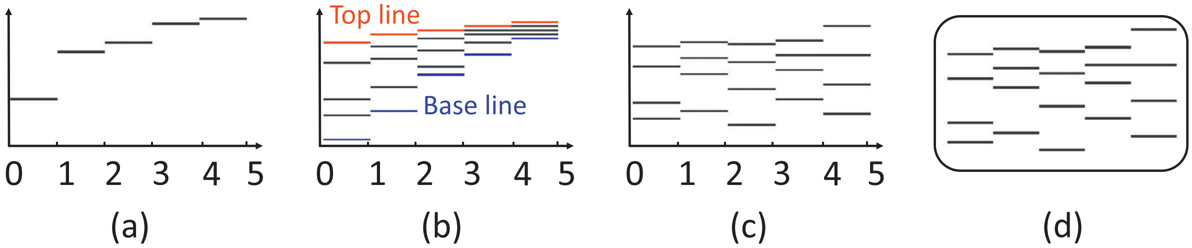

Fig. 2. Visual design evolves from (a) a basic stair-shape line, which represents certain graph characteristic distribution generated from one sample, to (b) stair-shape chart that all sample lines are put together in the same coordinates. Top-line and base-line are then generated for further normalization (c). (d) Normalized stair-shape lines are presented without coordinates since the research only needs relative comparison. The y-axis in (a) and (b) represents the value of characteristic distribution (node degree or cluster coefficient) that ranges from 0 to 1. In (c) and (d), the y axis represents the relative difference of values between each stair-shape line and base line. The x-axis keeps consistent from (a) to (d).

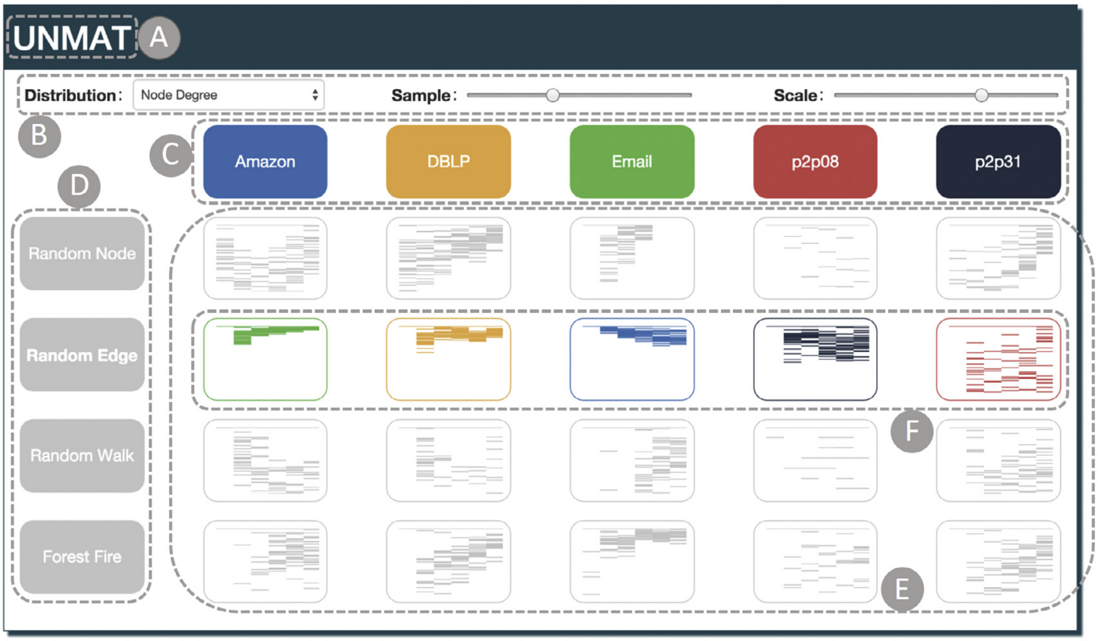

Fig. 3. Uncertainty comparison of four sampling approaches and five datasets in terms of node degree distribution. (A) The title of our visualization system UNMAT. (B) Control panel for specifying input parameters. (C) Five datasets acquired from Standford Large Network Collection. (D) Four sampling approaches for comparison and exploration. (E) Spreadsheet-style visualization for comparing different sampling approaches on various contexts. (F) Ranking stair-shape charts according to uncertainty information.

of multiple objects at the same time. Fig. 3 shows an example of the visualization.

System Overview UNMAT is an abbreviation of uncertainty matrix which inspires the design of spreadsheet-style visualization. The spreadsheet-style visualization employs matrix view that contains 4 rows and 5 columns corresponding to four graph sampling approaches and five larger graph datasets. Fig. 3(D) presents four column blocks which corresponds to four sampling approaches, random node, random edge, random walk as well as forest fire. These column blocks will be colored by blue, yellow, green and red separately when users want to compare different methods. Otherwise, they remains gray. Five colorful blocks are presented as blue, yellow, green, red and dark corresponding to Amazon, DBLP, Email, p2p08 and p2p31 datasets which comes from different contexts. All stair-shape charts are embed into the matrix view which helps users to compare them from two perspectives. Users can not only compare different graph sampling approaches with certain context but also discover which context the method match best. Only ranked stair-shape charts are colored according to their approaches or datasets while other charts still maintain gray. Additionally, users are limited to rank one column or row charts at a time. These visualization design prevents users from visual clutter. Our platform (UNMAT) also integrates a suit of interaction tools containing one extensible menu labeled distribution as well as two scrolling buttons labeled sample and scale respectively.

Interaction The extensible menu labeled distribution provides two choices of graph characteristics, namely, node degree distribution and clustering coefficient distribution. Users are allowed to select suitable graph characteristic to compare different sampling methods in various context. Two scrolling buttons labeled sample and scale separately are integrated in our platform. Sample button helps users to control the number of graph samples generated from original graph. The differences among stair-shape lines in one stair-shape chart are adjustable and accessible by scale button. The more larger the scale value is, the more dispersive the stair-shape chart looks like. Hence, it enable users to zoom in or out as they want. Once parameter space displayed in Fig. 3(B) is specified, all stair-shape charts can be rendered with gray color immediately. When users want to discover which dataset does a sampling method match best, they are supposed to click column blocks (as shown in Fig. 3(D)) labeled by the name of sampling approaches. After that, all stair-shape charts next to the clicked block are ranked by ascending order of VSM values and colored by their corresponding contexts. Meanwhile, all column blocks turn their color into gray in order to prevent users getting confused. Simi-

Table 1
Dataset description.

Dataset	Type	Nodes	Edges
Amazon	Directed	400,272	3,200,440
DBLP	Undirected	317,080	1,049,866
Email	Directed	265,214	420,045
p2p08	Directed	6301	20,777
p2p31	Directed	62,586	147,892

larly, users can compare various sampling methods by clicking row blocks (as shown in Fig. 3(C)).

6. Experiments result

We have conducted two experiments to evaluate different graph sampling methods in analysis of large networks from the aspect of uncertainty with our visual analytics framework and platform (UNMAT). Experiment I based on node degree distribution aims to compare different approaches on various datasets, while experiment II aims at discovering which context does a sampling method match better. Similarly, clustering coefficient distribution can also be utilized to help us gain deep insights of uncertainty information as node degree distribution do. Hence, we present results of experiment II on both node degree and clustering coefficient distribution. We simply introduce the datasets before presenting experiments result.

6.1. Datasets

In our research, we select five datasets, namely, amazon, dblp, email, p2p08 as well as p2p31 from Stanford Large Network Dataset Collection [30] and store them in a local database for further processing. They stand for various life occasions and help users to understand uncertainty information from different sampling approaches on various contexts. Amazon extracted on March 12, 2003 is a product co-purchasing network from Amazon.com[2]. DBLP is an unique undirected researchers collaboration network which helps us validate whether UNMAT can handling both directed and undirected networks or not. Email represents a network from a EU research institution. Finally, p2p08 collected on March 02 2002 is an Internet peer to peer network with a small scale while p2p31 collected on March 31, 2002 represents a much larger network. They can help users to validate whether a sampling method is scalable or not. Table 1 introduces more details including nodes and edges of selected datasets.

6.2. Experiment I

We conduct experiment I to compare and select most appropriate sampling approach in a specific context. Spreadsheet-style visualization accelerates the process of exploration in understanding uncertainty information from large graph sampling approaches. With accessible interaction tools, we adjust scrolling buttons to limit the number of samples into 25 and choose suitable scale. What's more, we select node degree distribution as the graph characteristic to be looked into. Parameter space is clearly shown in Fig. 3(B). We click row blocks one by one to generate a series of visualization outputs that ranked stair-shape charts. With these intuitive outputs, we can easily compare different sampling approaches on certain contexts.

As shown in Fig. 4(a), the four stair-shape charts are colored by yellow, blue, red and green which correspond to their sampling approaches. Random edge method has best performance compared to other approaches when it is utilized on amazon dataset. Random node method follows it closely and stay in the 2nd place. Although forest fire algorithm is more advanced and complexed than others, it just ranks 3rd in the name list. Random walk ranks the last one due to its larger variance of node degree distribution. Hence, we simply draw a conclusion that random edge is the most appropriate method while forest fire is the worst choice on amazon dataset. What's more, random node behaves slightly better than random walk method.

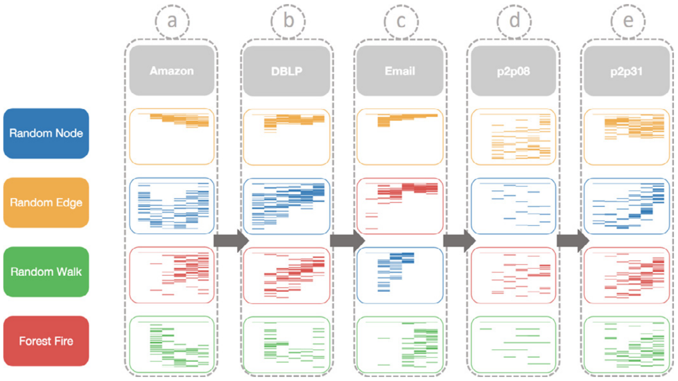

Fig. 4. Experiment I based on node degree distribution aims at comparing sampling approaches on various contexts: (a) Co-purchasing network from Amazon (b) DBLP collaboration network; (c) Email network from a EU research institution; (d) Internet peer to peer network (small); (e) Internet peer to peer network (large).

Fig. 4 depicts an overall picture of visual comparison consequences. Random edge still plays a leading role in the next four contexts while random walk stays in the 4th place all the time. Random node and forest fire only exchange their ranking position on Email dataset. In most contexts, random node still has a better performance than forest fire. We extend the simple aforementioned conclusion into an overall summarization.

Random edge method produce less uncertainty than other methods when it is implemented to simplify a large graph. However, forest fire algorithm always produces much more uncertainty than others. Hence, people should be careful about it when they utilize it to simplify a graph. Compared to random walk, random node method behaves better in most contexts. Due to a lack of knowledge about graph structure in advance, people should consider an order of sampling approaches as random edge, random node, random walk and forest fire.

6.3. Experiment II

In order to validate that both node degree and clustering coefficient distribution are useful in exploring uncertainty information from graph sampling, we conduct experiment II. Experiment II aims at discovering which contexts does a specific method match best and whether the method is scalable or not. In the extensible menu, we select node degree and clustering coefficient distribution separately. We also maintain the number of samples and scale values as what we do in Experiment I. After clicking row blocks separately, we are supposed to compare different datasets with certain sampling method. Hence, we can answer which contexts do sampling methods match best.

We summarize visualization, as shown in Fig. 5(a)–(d), and introduce our conclusions based on node degree distribution first. In Fig. 5(a), by applying random node method on the five datasets one by one, we conclude that DBLP (collaboration network) is the most appropriate context for random node method. What interests us is the different performances of random node between p2p08 and p2p31 datasets which comes from the same source with different scales. p2p31 ranks 2nd places while p2p08 is the last one.

[2] http://amazon.com.

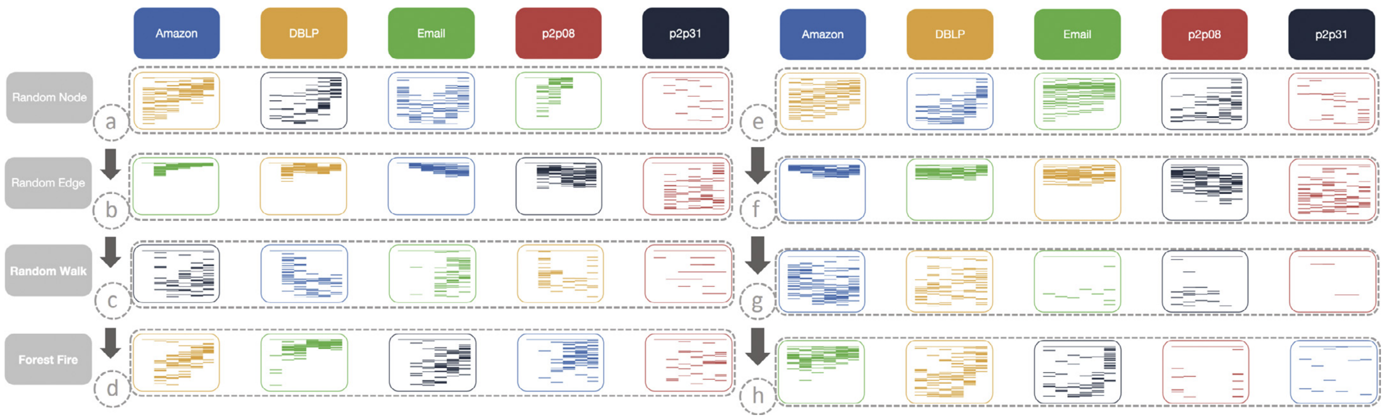

Fig. 5. Experiment II aims at discovering which context does a sampling method behave best and whether the method is scalable or not. All stair-shape charts are colored according to their contexts that are Amazon(blue), DBLP(yellow), Email(green), p2p08(red) as well as p2p31(black). Part (a) to (d) are based on node degree distribution while Part (e) to (h) are based on clustering coefficient. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The only difference between them is that the number of nodes and edges contained by p2p31 are almost ten times the number of nodes and edges in p2p08. We can draw a reliable conclusion that random node method is not scalable. However, random edge method has completely different conclusions compare to random node method.

Random edge method matches for email dataset best since the green stair-shape chart takes 1st place in the list. What's more, the difference between black stair-shape chart ranking 4th and red stair-shape chart ranking 5th can be ignorable. Hence, we can conclude that random edge method is scalable with different scales of graphs. Fig. 5(b) validates aforementioned analysis with intuitive visualization. We briefly summarize analysis consequences of random walk and forest fire. In Fig. 5(c) and (d), random walk matches p2p31 dataset best and is not scalable with graph scales while forest fire matches DBLP dataset best but not scalable too.

Considering another graph characteristic (e.g clustering coefficient distribution), we find conclusions summarized above are inconsistent with visualization shown in Fig. 5(e)–(h). Random node still suits DBLP dataset best while random edge and random walk have changed their suitable contexts from Email and p2p31 separately to Amazon dataset. Forest fire also suits another dataset (Email) when it comes to clustering coefficient distribution. Finally, being different from node degree, all methods are scalable when they are utilized to calculate clustering coefficient. Hence, node degree and clustering coefficient can help us to generate overall conclusions from different perspectives.

We validate the effectiveness of our visual analytics framework and platform (UNMAT) through experiment I and II. These experiment results indicate that random edge method has excellent performance in most contexts and is also scalable with different graph scales. Random walk method produces much more uncertainty than other methods when being utilized to simplify graphs. Additionally, it is not scalable with different graph scales. Random node and forest fire have their own weakness and strength. Hence, people should choose an appropriate one according to a specific context. We also suggest that it is better to implement random edge algorithm for graph sampling when we are lack of knowledge in advance.

6.4. User study

Our framework not only employs VSM score to quantify uncertainty of algorithm but also enable users to compare these sampling algorithms by eyes. As described previously, users can compare different sampling methods via stair-shape charts. The more dispersive a stair-shape chart looks like, the more uncertainty it contains. We have invited 16 volunteering users to attend (5 PhD students and 11 undergraduate students) the study. All users are familiar with basic visualization knowledge but not familiar with graph sampling. Hence, their judgment will not be disrupted by previous experiences. Each of them is asked to sort different sampling algorithms on real world (discussed in Section 6.1). All surveys are completed by users independently. During research, we ask each user to score sampling algorithms in each dataset from 1 point to 4 point. 1 point indicates that an algorithm produces much more uncertainty compared to other algorithms. 4 point indicates that an algorithm produces fewer uncertainty than others. Users must assign algorithms a unique score. Then we sort algorithms from highest points to lowest according to their average scores. Table 2 presents the result of our user study. The order sorted by users is listed in the first column on certain dataset. The result, as shown in the Table 2, indicates that both variance score method (VSM) and stair-shape charts keep consistent conclusions in most cases. Thus, we conclude that VSM can assist users to compare sampling algorithms.

Table 2
Algorithm sorting.

Method	Amazon		DBLP		Email		p2p08		p2p31	
Node	2	2	2	2	1	3	2	2	2	2
Edge	1	1	1	1	2	1	3	1	4	1
Walk	3	4	4	4	2	2	4	4	2	4
Fire	4	3	2	3	4	4	1	3	1	3

7. Discussion

In our research, we have selected five network datasets from Stanford Large Network Dataset Collection [30] and implement four classical graph sampling approaches on datasets, which can help us to facilitate comprehensive analysis and draw reliable conclusions. However, it still remains unknown whether UNMAT can be utilized in extreme-scale network with billions node and edges such as social network on Twitter. With the growing needs of exploring extreme-scale networks, we considere to employ parallel computing techniques to accelerate the performance of our framework and support real-time comparison of graph sampling algorithms. Overall, our work provides a convenient means for a non-expert user to evaluate different sampling algorithms for a certain application and dataset.

Uncertainty is an important issue in graph sampling field and can affect the reliability of graph analysis. Considering that visualization combines human's cognition and machine's computation power, we have developed a set of visualization tools instead of simply presenting statistics for users. Stair-shape chart and spreadsheet-style visualization are developed to assist users comparing different sampling approaches from uncertainty information. However, we only employs two graph characteristics, namely node degree and cluster coefficient, in this work. Integrating other characteristics into our framework is a non-trivial issue because different characteristics have different mathematical forms. Thus, we plan to figure out this issue in the future work.

Stair-shape chart is efficient for users to understand uncertainty information from graph sampling which is validated by our experiments. However, this visualization is really dependent on the number of displayed samples. On the one hand, with less graph samples, stair-shape chart is so sparsely that human cannot gain any insights from several lines. On the other hand, with more graph samples, stair-shape chart looks more continuous which hides users to gain kernel density information. Hence, we are considering employing kernel density estimation method to generate more smooth visualization while keep insights from graph samples. More efforts will be spent on aforementioned issues to improve our visual analytics framework and platform.

8. Conclusion

This paper investigates the uncertainty problem in graph analysis process. We present a visual analytics framework and a platform called UNMAT to support exploration of uncertainty information from various graph sampling approaches. With the introduction of spreadsheet-style visualization, users can compare different sampling methods on various contexts at the same time. Integrating variance score method (VSM) and stair-shape chart also empower the ability of visual comparison of UNMAT. Moreover, a set of interaction tools are provided for generating reliable conclusions by incorporating human's cognitive power and machine's computation. Finally, we will spend more efforts on dealing with extreme-scale large dataset, such as social media network, and supporting real-time interactions.

References

[1] D. Rafiei, S. Curial, Effectively visualizing large networks through sampling, in: Proceedings of IEEE Visualization, 2005, pp. 375–382.
[2] Y.-Y. Ahn, S. Han, H. Kwak, S. Moon, H. Jeong, Analysis of topological characteristics of huge online social networking services, in: Proceedings of the International Conference on World Wide Web, 2007, pp. 835–844.
[3] L. Lovász, Random walks on graphs: a survey, Combinatorics 2 (1) (1993) 1–46.
[4] A. Mislove, M. Marcon, K.P. Gummadi, P. Druschel, B. Bhattacharjee, Measurement and analysis of online social networks, in: Proceedings of the ACM Internet Measurement Conference (SIGCOMM), 2007, pp. 29–42.
[5] J. Leskovec, C. Faloutsos, Sampling from large graphs, in: Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining SIGKDD, 2006, pp. 631–636.
[6] T. Wang, Y.C.Z. Zhang, T. Xu, L. Jin, P. Hui, B. Deng, X. Li, Understanding graph sampling algorithms for social network analysis, in: Proceedings of the International Conference on Distributed Computing Systems Workshops, 2011, pp. 123–128.
[7] Y. Wu, N. Cao, D. Archambault, Q. Shen, H. Qu, W. Cui, Evaluation of graph sampling: a visualization perspective, IEEE Trans. Visual. Comput. Graphics 23 (1) (2017) 401–410.
[8] M.H. Ramsey, Sampling as a source of measurement uncertainty: techniques for quantification and comparison with analytical sources, J. Anal. At. Spectrom. 13 (1998) 97–104.
[9] S.E. Schaeffer, Graph clustering, Comput. Sci. Rev. 1 (1) (2007) 27–64.
[10] M. Gjoka, M. Kurant, C.T. Butts, A. Markopoulou, Walking in facebook: a case study of unbiased sampling of OSNs, in: Proceedings of the IEEE International Conference on Computer Communications (INFOCOM), 2010, pp. 1–9.
[11] B. Ribeiro, D. Towsley, Estimating and sampling graphs with multidimensional random walks, in: Proceedings of the ACM Internet Measurement Conference (SIGCOMM), 2010, pp. 390–403.
[12] T. Wang, Y. Chen, Z. Zhang, P. Sun, B. Deng, X. Li, Unbiased sampling in directed social graph, ACM SIGCOMM Comput. Commun. Rev. 40 (4) (2010) 401–402.
[13] D. Xiu, Numerical methods for stochastic computations: a Spectral method approach, Princeton University Press, 2010.
[14] M. Kleiber, T.D. Hien, The stochastic finite element method: basic perturbation technique and computer implementation, 1st, John Wiley & Sons, 1993.
[15] C.D. Correa, Y.-H. Chan, K.-L. Ma, A framework for uncertainty-aware visual analytics, in: Proceedings of IEEE Symposium on Visual Analytics Science and Technology, 2009, pp. 51–58.
[16] W.K. Liu, T. Belytschko, A. Mani, Random field finite elements, Int. J. Numer. Methods Eng. 23 (10) (1986) 1831–1845.
[17] F. Yamazaki, M. Shinozuka, G. Dasgupta, Neumann expansion for stochastic finite-element analysis, J. Eng. Mech. 114 (8) (1988) 1335–1354.
[18] D. Xiu, J.S. Hesthaven, High-order collocation methods for differential equations with random inputs, SIAM J. Scient. Comput. 27 (3) (2005) 1118–1139.
[19] A.T. Pang, C.M. Wittenbrink, S.K. Lodha, Approaches to uncertainty visualization, Vis. Comput. 13 (8) (1996) 370–390.
[20] C.R. Johnson, A.R. Sanderson, A next step: visualizing errors and uncertainty, IEEE Comput. Graph. Appl. 23 (5) (2003) 6–10.
[21] C. Olston, J.D. Mackinlay, Visualizing data with bounded uncertainty, in: Proceedings of the IEEE Symposium on Information visualization, 2002, pp. 37–40.
[22] S. Deitrick, R. Edsall, The influence of uncertainty visualization on decision making: an empirical evaluation, in: Progress in Spatial Data Handling, Springer Berlin Heidelberg, 2006, pp. 719–738.
[23] J. Thomson, B. Hetzlera, A. MacEachrenb, M. Gaheganb, M. Pavel, A typology for visualizing uncertainty, in: Proceedings of Visualization and Data Analysis, 2005, pp. 146–157.
[24] T. Zuk, S. Carpendale, Visualization of uncertainty and reasoning, in: Proceedings of the International Symposium on Smart Graphics, 2007, pp. 164–177.
[25] Y. Wu, G.-X. Yuan, K.-L. Ma, Visualizing flow of uncertainty through analytical processes, IEEE Trans. Visual. Comput. Graphics 18 (12) (2012) 2526–2535.
[26] D. Feng, L. Kwock, Y. Lee, R. Taylor, Matching visual saliency to confidence in plots of uncertain data, IEEE Trans. Visual. Comput. Graphics 16 (6) (2010) 980–989.
[27] K. Potter, J. Kniss, R. Riesenfeld, C. Johnson, Visualizing summary statistics and uncertainty, Comput. Graphics Forum 29 (3) (2010) 823–832.
[28] J. Sanyal, S. Zhang, G. Bhattacharya, P. Amburn, R. Moorhead, A user study to compare four uncertainty visualization methods for 1d and 2d datasets, IEEE Trans. Visual. Comput. Graphics 15 (6) (2009) 1209–1218.
[29] T. Zuk, S. Carpendale, Theoretical analysis of uncertainty visualizations, in: Proceedings of Visualization and Data Analysis, 2006, pp. 66–79.
[30] J. Leskovec, A. Krevl, SNAP Datasets: Stanford large network dataset collection, 2014, (http://snap.stanford.edu/data).
[31] J. Leskovec, R. Sosič, Snap.py: SNAP for Python, a general purpose network analysis and graph mining tool in Python, 2014, (http://snap.stanford.edu/snappy).