Knowledge-Based Systems 212 (2021) 106620

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

A dynamic sampling algorithm based on learning automata for stochastic trust networks

Mina Ghavipour [a,b,*], Mohammad Reza Meybodi [b]

[a] School of Computer Science, Institute for Research in Fundamental Sciences (IPM), Tehran, Iran
[b] Department of Computer Engineering and IT, Amirkabir University of Technology, Tehran, Iran

ARTICLE INFO

Article history:
Received 13 June 2020
Received in revised form 2 November 2020
Accepted 21 November 2020
Available online 26 November 2020

Keywords:
Trust networks
Stochastic graphs
Graph measures
Network sampling
Learning automata

ABSTRACT

Trust is known as an important social concept and an effective factor in all human interactions in social networks. Users tend to interact with trusted people with whom they have had positive experiences. Trust is updated over time as a result of these repeated interactions. Even though dynamicity is a universally accepted property of the social trust, trust networks are often modeled as static digraphs. In this paper, we first propose that a stochastic graph model, where the weights associated with edges are random variables with unknown distributions, may be a better candidate for representing trust networks. Then, we review the literature on analyzing complex networks and determine graph measures which are most appropriate with respect to the special properties of the concept of trust. Considering trust-specific measures, we finally propose a dynamic algorithm for sampling from stochastic trust networks, which is an extension of Frontier sampling. Even though there exist a few sampling methods which address edge weights and their variations over time through the sampling process, these methods are unable to accurately preserve the properties of trust networks. The proposed algorithm in this paper uses learning automata to tackle the disconnectivity problem of sampled subgraphs by Frontier sampling and, at the same time, capture the changes of edge weights through the sampling process. Our experimental results on the well-known trust network datasets indicate that the proposed sampling algorithm preserves more accurately the trust-specific measures of trust networks compared to existing sampling methods.

1. Introduction

Social networks have become popular as a medium for disseminating information and interacting with others. The public accessibility of these networks with the ability to share information, opinions, and experience offers great promise to companies and governments. Social networks are used by companies and governments to deliver their services to customers and citizens. These networks are intrinsically non-deterministic, and topological structures and human behaviors in them have time-varying nature. As a result, using a static graph model with fixed weights for social networks is too simplistic and restrictive for many real-world network problems. Authors in [1] were the first to address the dynamicity of edge weights in social networks and propose using stochastic graphs, in which edge weights are random variables with unknown distributions, for modeling and analyzing these networks. They presented new generalizations of some graph measures for stochastic networks and proposed several learning automata-based algorithms for calculating these measures. Very recently, some research works have been dedicated to solving various problems on stochastic social networks [2–5].

Trust is considered a vital factor in online social networks. The success of social interactions depends on the level of trust that users have with each other as well as with service providers. In this way, trust influences the quality of interactions in social networks and reduces uncertainties and risks from unreliable users. From the other side, the experience of users in their interactions also affects the level of trust between them in such a way that trust increases between users if the experience is positive and decreases otherwise. As a result, dynamicity is an important property of social trust.

Various techniques have been used in literature for addressing the dynamicity property of the trust, for example, using temporal window [6–10], aging of old interactions [11–13], giving more weight to recent interactions [14–16], periodically computing the value of trust [17,18] and considering only the most recent interaction [19,20]. However, even though the dynamic nature of trust has been widely accepted in the literature, most works in this scope have been modeled trust networks as static digraphs and developed deterministic algorithms for solving problems on

* Corresponding author at: School of Computer Science, Institute for Research in Fundamental Sciences (IPM), Tehran, Iran.
E-mail address: mina_ghavipour@aut.ac.ir (M. Ghavipour).

https://doi.org/10.1016/j.knosys.2020.106620
0950-7051/

these networks [21–36]. In this paper, we propose that a stochastic graph can better represent trust networks with respect to the dynamicity property of the trust, and therefore it is a more accurate model for these networks. To the best of our knowledge, we are the first to introduce the concept of a stochastic trust network as a model for trust networks.

Since online social networks have gained more popularity among internet users in recent years, computer scientists and sociologists have started to study and analyze the characteristics of these networks. Social network analysis (SNA) as an inherently interdisciplinary field focuses on social relations between users rather in the users themselves. The main purpose of SNA is to study both contents and patterns of relations in online social networks to understand the implications of these social relations. Trust networks are of the most important directed and weighted social networks. There exist different generalizations of the original graph measures proposed in the literature, which in addition to considering the topological structure, also address directionality and weights of connections. However, since these generalizations have been designed with different types of networks in mind, despite having the same goal they behave quite differently from each other. As a result, when studying a network of a special type, the first and most important challenge is that which definitions must be adopted among several existing various definitions. To tackle the challenge of selecting measures, we survey the literature for the existing graph measure generalizations, and choose the most appropriate of them with respect to the properties of trust. Wherever it is necessary, we also propose new definitions of measures, considering the topological structure, the directionality, and the trust weights assigned to connections in a trust network.

In addition to the challenge of selecting graph measures, there exist many other factors that make it difficult, if not impossible, to study social networks in their entirety. First and foremost, the huge size of many real-world networks makes it computationally infeasible to study the entire network. Moreover, some online networks are not completely visible to the public (e.g., Facebook) or can be accessed only through crawling (e.g., Web). In other cases, the size of networks may not be as large but the measurements required to observe the underlying networks are costly. As a result, network sampling is at the heart and foundation of the study for understanding network structure.

The main goal of network sampling is to produce representative subgraphs with a smaller size which can be used to make inferences about the full graph. Many network sampling methods have been developed in the literature [37–44]. However, these methods attempt to only match the structural properties of the original graph and in the case of weighted graphs, they ignore edge weights through their sampling process. This is while many social networks include weights associated with their edges to differentiate the edges in terms of their strength or capacity [45]. Moreover, the edge weights in many of these networks change over time, indicating the non-deterministic nature of these networks. Authors in [1] proposed that stochastic graphs, in which edge weights are random variables with unknown distributions, can better represent social networks. Very recently, some research works have been dedicated to sampling subgraphs from weighted [46] and stochastic [3] social networks.

In this paper, we also develop a dynamic sampling algorithm based on learning automata for stochastic trust networks, whose sampled subgraphs can accurately preserve our trust-specific measures. The proposed algorithm, called D-LAFS, is an extension of the Frontier sampling algorithm (m-dimensional random walk sampling) [43] which uses m dependent random walkers. It has been shown that sampling methods based on random walks performs best for preserving the properties of the original graph [47]. However, in the presence of disconnected or loosely connected original graphs, Random walks can suffer from large estimation errors [43]. Therefore, authors in [43] proposed a new m-dimensional random walk sampling method and indicated that this method, called Frontier sampling, exhibits lower estimation errors while preserving all of the important sampling properties of a regular random walk.

The problem with Frontier sampling is that this method, because of using m random walkers, is probable to sample subgraphs with multiple disconnected components. Moreover, Frontier sampling does not take into consideration edge weights and thus their changes over time. To tackle these problems, the sampling algorithm D-LAFS proposed in this paper uses learning automata to guarantee the connectivity of sampled subgraphs and, at the same time, perform smart sampling for minimizing the number of samples taken from the edge weights of the original graph. In D-LAFS, learning automata are biased to high degree nodes and learn to sample connected subgraphs. Learning automata has been successfully used in the literature for graph sampling [37,38].

To evaluate the performance of the proposed sampling algorithm, we conduct several experiments on the well-known trust network datasets. Our experimental results indicate that the proposed algorithm more accurately preserves trust-specific statistic generalizations of the original graph comparing to the dynamic version of existing sampling algorithms including Random Walk sampling and Frontier sampling, as well as stochastic sampling algorithm eDLAS-SG [3].

The rest of the paper is organized as follows. The next section provides background and related work on social trust, social network analysis, graph sampling, and learning automata. In Sections 3 and 4, we respectively describe the proposed stochastic digraph model for trust networks and the generalizations of graph measures, namely degree, clustering coefficient, and path length. Section 5 describes the proposed dynamic algorithm for sampling from stochastic trust networks. The experimental results on well-known trust network datasets are given in Section 6. Finally, the last section discusses and concludes the paper.

2. Background and related work

In this section, we state the importance of trust in social networks and discuss the dynamic nature of trust and existing techniques for modeling this dynamicity. We also review the literature on social network analysis, graph measures, and graph sampling. At last, a background of learning automata is briefly presented.

2.1. Social trust in online social networks

Trust is widely accepted as an important component of human social relationships which affects the decision-making of users in online social networks [48]. Web-based social networks have provided an appropriate infrastructure for disseminating information and connecting like-minded people. Since the success of such activities relies on the level of trust between users, trust is known as an essential and important factor in a successful social network. A lot of social applications, such as Epinions (www.epinions.com) and Advogato (www.advogato.org), provide a web of trust for users to express their trust level in other people. Some of the most important properties of trust are [49]: subjective, propagative, non-transitive, asymmetric, and dynamic. We briefly describe these trust properties in the following paragraphs.

Subjective. The subjective property of trust refers to that if user A trusts user B, this does not necessarily mean that B also will

trust A [50]. Mui et al. [51] argued that a user is likely to have different levels of trust in the eyes of other users.

Propagative. Trust is propagative, namely if A trusts B, who in turn trusts C, A can derive some amount of trust on C based on the value of her trust in B and the value of B's trust in C. Based on this property, trust information can be propagated from one user to another in a social network, creating chains of trust [50–53].

Non-transitive. In general, trust is not transitive. If A trusts B and B trusts C, this does not necessarily imply that A trusts C [54,55]. Sometimes, the propagative nature of trust is confused with its transitive nature in the literature.

Asymmetric. The asymmetric nature of trust refers to that user A may trust user B more than he is trusted back [56]. This asymmetry occurs because of differences in peoples' opinions, expectations, and beliefs.

Dynamic. Trust is built up over time as a result of repeated direct interactions between users. It can increase or decrease with new interactions [57]. It may also decay with time. The experiences of new interactions are more important than those of old ones, since the experiences of old interactions may become obsolete or irrelevant with time [49]. This property refers to the dynamic nature of trust. In general, there are two approaches for updating the value of trust [58]: event-driven and time-driven. In the event-driven approach, the trust value between two users is updated after an interaction or event is made. The various techniques have been proposed in the literature considering this approach, for example, aging of old interactions [11–13], giving more weight to recent interactions [14–16], and considering only the most recent interaction [19,20]. In contrast, in the time-driven approach, the experiences of interactions are collected periodically and the trust value is updated by applying a trust aggregation technique. In the case of no new experiences over a period of time, trust decay is applied as recent information should be trusted more than past information. Some examples of various techniques proposed in the literature considering this approach include: using a temporal window [6–9] and periodically computing the value of trust [17,18]. Authors in [6,7] presented the technique PeerTrust which allows users to choose a temporal window for dynamic aging of old interactions as per their need. The technique PowerTrust proposed in [17] computes trust values periodically to ensure that the computed values are up-to-date.

Despite the dynamic nature of trust has been widely accepted in the literature, most works in this scope have been modeled trust networks as static digraphs and developed deterministic algorithms for solving problems on these networks [21–36]. In this paper, we propose that a stochastic graph model can represent better the dynamicity property of the trust in trust networks and therefore it is a more appropriate model for these networks.

2.2. Online social network analysis

In recent years, as online social networks have been gaining more popularity among internet users, sociologists and computer scientists have started to investigate and analyze the characteristics of these networks. Social network analysis (SNA) focuses on social relations established between users rather in the users themselves. The main goal of SNA is to examine both contents and patterns of relations in social networks to understand the implications of these social relations. A substantial body of recent research has been dedicated to investigating the topological properties of social networks [59–63].

Many social networks display a large heterogeneity in the strength of social relations [45]. Besides, the relationships in many of these networks are directed and asymmetric [64]. Since the graph measures do not consider the strength and directionality of connections, there have been several attempts in the literature to extend the graph measures to the case of weighted [45, 65–68] and (or) directed networks [64,69–71]. However, since measure generalizations have been designed with different types of networks in mind (for example, airport and collaboration networks [45], stock correlation networks [72], and gene co-expression networks [73]), the questions behind them are different. Therefore, even though various generalizations of a measure have the same goal, they behave quite differently from each other [74] and there is no single general-purpose metric for computing any of the measures in directed and weighted complex networks [75]. As a result, when analyzing a network of a special type, the first and the most important challenge is that among several existing definitions for each graph measure, which definition must be adopted.

2.3. Graph measures

A network can be described by a graph $G(V, E)$, where V is a set of N nodes, $v_i \in V$, that are connected by a set of M links, $e_{ij} \in E$. To formally characterize a binary undirected graph, it is sufficient to provide its adjacency matrix, i.e. a symmetric $N \times N$ binary matrix A whose entry $a_{ij} = a_{ji} = 1$ if and only if a link connecting nodes v_i and v_j exists, and zero otherwise. In contrast, a weighted undirected graph is represented by a symmetric $N \times N$ weight matrix W, whose entry $w_{ij} = w_{ji} > 0$ measures the strength of the connection between node v_i and v_j, and it is zero if no link exists between the two nodes. In the case of a directed graph, links are oriented and social relationships are not necessarily symmetric, which implies a non-symmetric adjacency or weight matrix. These matrices can be used to derive various graph measures, including the node degree, the clustering coefficient, and the shortest path length.

In what follows, we discuss the existing literature on these measures in different types of complex networks.

2.3.1. Degree

The degree of a node v_i, denoted as k_i, is a measure of the immediate adjacency and the involvement of v_i in the network. Despite its simplicity, the node degree is known as a natural measure to assess the importance and centrality of a user in a social network [76]. The main limitation of this measure is that it does not consider the global structure of the network [77]. For a binary undirected network, k_i is computed as the number of links incident on v_i or, similarly, as the number of neighbors of the node v_i [78], i.e. $k_i = \sum_{j \neq i} a_{ij}$. When analyzing a weighted undirected network, the node degree is generally extended to the node strength, which for a node v_i is defined as the sum of weights of all links incident on v_i [45,66,67,79–81]. This measure is formalized as $s_i = \sum_{j \neq i} w_{ij}$. This definition is equal to the definition of node degree if the network is binary, i.e. each link has a weight of 1. It has been shown that there is a correlation between the strength of nodes and their degree such that the average strength of nodes with the same degree increases with the degree [45].

For a binary directed network, there are two variants of node degree: *in-degree* and *out-degree*. For a node v_i, the former is computed as the number of links that are directed towards v_i, i.e. $k_i^{in} = \sum_{j \neq i} a_{ji}$, and the latter as the number of links that originate from v_i, i.e. $k_i^{out} = \sum_{j \neq i} a_{ij}$ [82]. The total degree of a node v_i is the sum of its *in*- and *out*-degree. Considering the definitions of k_i^{in} and k_i^{out}, the measures of *in-strength* and *out-strength* for a node v_i in a weighted directed network can be defined as $s_i^{in} = \sum_{j \neq i} w_{ji}$ and $s_i^{out} = \sum_{j \neq i} w_{ij}$, respectively.

Since the node strength does not take into consideration the number of nodes adjacent to a given node v_i, authors in [77] proposed a new measure of degree centrality for weighted networks, which combines both degree and strength of the node v_i.

$$s_i^w = k_i \times \left(\frac{s_i}{k_i}\right)^{\alpha} \tag{1}$$

where α denotes a positive tuning parameter, which determines the relative importance of the number of links compared to weights. For $\alpha < 1$, both increments in the number of links and the link weights will increase the value of the measure, and conversely for $\alpha > 1$, a node with on average stronger links will get a higher degree. The variants of Opsahl's definition for weighted directed networks are as follows:

$$s_i^{w-in} = k_i^{in} \times \left(\frac{s_i^{in}}{k_i^{in}}\right)^{\alpha} \tag{2}$$

$$s_i^{w-out} = k_i^{out} \times \left(\frac{s_i^{out}}{k_i^{out}}\right)^{\alpha} \tag{3}$$

2.3.2. Clustering coefficient

The clustering property of a network refers to that if there are three nodes v_i, v_j and v_h in the network such that v_i is connected to v_j and v_h, the likelihood that v_j and v_h are also connected tends to be greater than the probability of a connection randomly created between two nodes [83–90]. The mechanisms that are responsible for this increase in the probability that two users will be connected when they share a common acquaintance have been investigated for social networks [91–93]. Two main measures for assessing the clustering tendency of nodes are the local clustering coefficient [90] and the global clustering coefficient [82,94–97]. The local clustering coefficient (LCC) is based on the local density and is computed as the fraction of the number of links connecting the neighbors of a user over the total number of possible links between them [90].

$$C_i = \frac{\sum_{j\neq i}\sum_{h\neq i,j} a_{ij}a_{jh}a_{ih}}{k_i(k_i-1)} \tag{4}$$

where numerator counts the number of closed triplets centered on the node v_i, namely a set of three nodes, including v_i, which are connected by three links and form a triangle. The coefficient takes values in the range of $[0, 1]$. In simple terms, C_i is the mean probability of being neighbor two nodes that are also the neighbors of the same another node. In a completely connected network, where all triplets are closed, C is equal to 1, whereas, in a classical random network, $C \to 0$ as the network size grows. More specifically, in a classical random network, C is equal to the probability that two randomly selected nodes are connected [82].

The local clustering coefficient does not take into consideration the weight and directionality of links connecting a node to its neighbors. Moreover, this coefficient is biased by correlations with node degrees: a node with a higher degree is likely to have a smaller local clustering than a node with a lower degree [98,99]. There are several attempts to generalize the local clustering coefficient to weighted networks [45,72,73,75,100–103]. Barrat et al. [45] were the first to propose a weighted clustering measure as

$$C_i^W = \frac{1}{s_i(k_i-1)}\sum_{j\neq i}\sum_{h\neq i,j}\frac{w_{ij}+w_{ih}}{2}a_{ij}a_{jh}a_{ih} \tag{5}$$

The authors were interested in scientific collaboration and air-transportation networks. Barrat's measure quantifies the local cohesiveness within the closed triplet centered at the node v_i. However, it considers only the average weight of the links incident on v_i within a triplet and neglects the weight w_{jh}. In contrast, Lopez-Fernandez et al. [100] presented a generalization of the clustering coefficient measure which utilizes only the weight of the link between the neighbors of the node v_i. Another version of this definition has been proposed in [104].

The problem with these two measures defined by Barrat and Lopez-Fernandez is that they do not capture the total connectivity within a closed triplet. The clustering measure defined by Onnela et al. in [72] for weighted networks addresses this issue and takes into consideration the weights of all the links forming a triplet. As shown in [72], this measure works well for stock correlation networks. However, it requires the availability of the degree information related to the underlying binary network. The binary degrees introduce a bias to the clustering coefficient computation that should be avoided. Zhang et al. [73] addressed this issue by proposing a weighted clustering measure for gene co-expression networks, which computes the clustering coefficient only through the use of link weights. Holme et al. [86] presented a similar measure, which also includes the possible maximum weight in the denominator. Another similar definition has also been proposed in [65,105], where the link weights are interpreted as probabilities such that in an ensemble network, the nodes v_i and v_j are connected with the probability of w_{ij}.

Although the measure defined by Zhang presents an alternative to Onnela's requirement to utilize binary degrees, it suffers false positive identification of triplets, which occurs when at least one of the links forming the closed triplet has a significantly lower weight than the other ones while the clustering coefficient is still large [75]. As a result of false identification, a "strongly" linked closed triplet is identified as equivalently "strong" as an open triplet. Bolanos et al. [80] proposed a weighted clustering coefficient definition to decrease the number of false-positive identifications by incorporating the total triplet intensities within the denominator, as given in Eq. (6). In their measure, they treat the link weights as probabilities, namely w_{ij} is the probability of information flowing through the link between v_i and v_j.

$$C_i^W = \frac{3\sum_{j\neq i}\sum_{h\neq i,j} w_{ij}w_{jh}w_{ih}}{\sum_{j\neq i}\sum_{h\neq i,j} w_{ij}w_{jh} + \sum_{j\neq i}\sum_{h\neq i,j} w_{ij}w_{jh}w_{ih}} \tag{6}$$

This definition computes the clustering coefficient as the proportion of the total probability of forming a closed triplet to the total probability of all possible (closed and open) triplets with respect to v_i, and guarantees its value to be between 0 and 1 using all of the link weights. In their proposed measure, authors consider an open triplet of the node v_i as one in which v_i is the starting point (or the endpoint), and not the center node, in a connected set of three nodes. In this way, they attempt to reduce the likelihood of false positives.

The literature also presents various attempts at developing a clustering coefficient measure designed for directed networks [64,106,107]. Fagiolo [64] has generalized the standard local clustering coefficient defined by Watts and Strogatz for binary and weighted directed networks. He first presents a general generalization of the clustering coefficient measure, in which all possible directed triplets are considered, no matter the directions of their links. For binary directed networks, the clustering measure is defined as the ratio between all directed triplets formed by the node v_i and the number of all possible triplets that could be formed. The extension of this definition to weighted directed networks can be obtained by replacing the number of directed triplets formed by v_i with its weighted counterpart.

Since these definitions treat all possible directed triplets as if they were the same, i.e. directions were irrelevant, they are not enough to characterize the richness of triplet patterns that take place in directed networks. In these networks, triplets with links pointing in different directions have different interpretations. Therefore, in [64] four more definitions have been proposed, considering four possible patterns of directed triplets (see Fig. 1). Herein, we only present the definition given for the *Middleman* pattern, which refers to when one of the neighbors of v_i holds two paths towards the other neighbor, one with a direct link and the

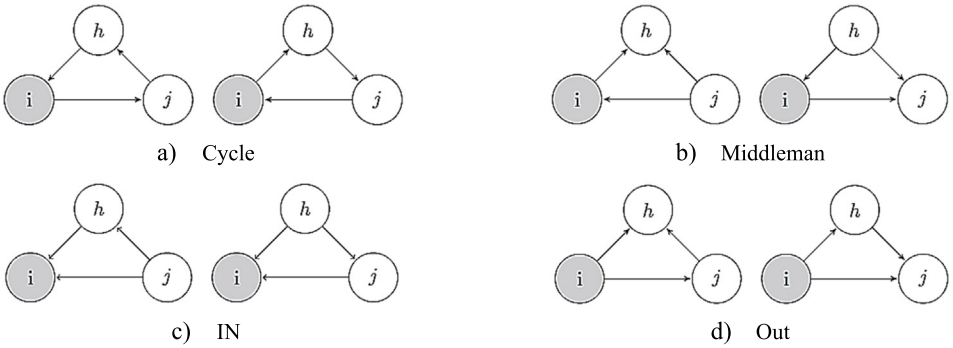

Fig. 1. A taxonomy of the patterns of directed triplets.

other using v_i as the intermediate node. In this case, the clustering coefficient for a binary network is computed as:

$$C_i^{D-mid} = \frac{\sum_{j\neq i}\sum_{h\neq i,j}(a_{ij}a_{hj}a_{hi} + a_{ih}a_{jh}a_{ji})}{2\left[k_i^{in}k_i^{out} - \sum_{j\neq i} a_{ij}a_{ji}\right]} \tag{7}$$

and for a weighted network is given as

$$C_i^{DW-mid} = \frac{\sum_{j\neq i}\sum_{h\neq i,j}\left(w_{ij}^{\frac{1}{3}}w_{hj}^{\frac{1}{3}}w_{hi}^{\frac{1}{3}} + w_{ih}^{\frac{1}{3}}w_{jh}^{\frac{1}{3}}w_{ji}^{\frac{1}{3}}\right)}{2\left[k_i^{in}k_i^{out} - \sum_{j\neq i} a_{ij}a_{ji}\right]} \tag{8}$$

When measuring the clustering coefficient in directed networks, it is important to account separately for each of the above patterns. Based on this, each measure is defined as the ratio between the number of triplets of a pattern formed by the node v_i and the total number of possible triplets of that pattern that v_i can form. Fagiolo's measures have been widely used for analyzing different directed complex networks [69–71].

Authors in [107] proposed a measure called transitivity for calculating closure in directed networks, which produces the same results as the global clustering coefficient if applied to undirected networks [82,94]. In their work, they considered a more sophisticated definition of a triplet. A triplet centered on the node v_i is defined as a chain of two directed links between three nodes such that v_i has one incoming and one outgoing link. Triplets that do not satisfy this condition are termed as vacuous. A non-vacuous triplet can be either transitive or intransitive. In a transitive triplet, there exists a link from the first node to the last node of the chain. In Fig. 1, the In and Out patterns are vacuous, the *Cycle* pattern is intransitive, and the *middleman* pattern is transitive. Considering this definition of transitive triplets, work in [106] presented a generalization of the global clustering coefficient for directed weighted networks.

2.3.3. Path length

One of the most important measures to analyze the properties of a network is the path length or the shortest path length between two nodes (popularly known as the number of "degrees of separation" in social networks [108]). A path is a sequence of nodes such that consecutive pairs of nodes are connected by a link. In binary networks, the path length from node v_i to node v_j, denoted as d_{ij}, is defined as the minimum number of links connecting v_i and v_j [88,107,109–111].

$$d_{ij} = \min_{p(i,j)\in\Gamma(i,j)}\left[\sum_{h,l\in p(i,j)} a_{hl}\right] \tag{9}$$

where $p(i,j)$ denotes a path from v_i towards v_j, $\Gamma(i,j)$ is the set of all possible paths between these two nodes, and clearly $d_{ij} = d_{ji}$ in the case of an undirected network. When analyzing the path length, an important assumption is that the intermediate nodes increase the cost of interactions.

There are several attempts in the literature to identify the shortest paths in different weighted networks [109–112]. Dijkstra [112] presented an algorithm to find the path of least resistance in networks where the link weights represent costs. For example, in the case of transportation networks, the link weight describes the physical distance and/or time involved in these networks, and it is inversely proportional to the strength of the corresponding connection [113,114]. In such networks, the weighted path length d_{ij}^w is defined as the minimum of the sum of the link weights along all possible paths between two nodes v_i and v_j.

$$d_{ij}^W = \min_{p(i,j)\in\Gamma(i,j)}\left[\sum_{h,l\in p(i,j)} w_{hl}\right] \tag{10}$$

However, in most weighted networks the weights are a representation of link strength and not the cost of them, and therefore they need to be reversed before applying shortest path algorithms in these networks [88,115]. For example, in the case of communication networks, the efficiency of a communication channel between two nodes is proportional to its weight. In such networks, the weighted path length d_{ij}^w is defined as the minimum of the sum of the inverse weights along all possible paths between two nodes v_i and v_j [116,117]. The same measure has been defined in [118] for computing the weighted path length in brain networks. To take into consideration both the link weight and the intermediate nodes, authors in [77] proposed a path length measure using a positive tuning parameter α.

$$d_{ij}^W = \min_{p(i,j)\in\Gamma(i,j)}\left[\sum_{h,l\in p(i,j)} \frac{1}{w_{hl}^{\alpha}}\right] \tag{11}$$

In this measure, for $\alpha < 1$, shorter paths with weak links are favored over longer paths with strong links, and conversely for $\alpha > 1$, the effect of intermediate nodes is relatively unimportant compared to the link weights and paths with a greater number of intermediate nodes are favored.

Bolanos et al. [80] have proposed a different definition in which the weighted path length between two nodes v_i and v_j is considered to be the maximum possible connection strength between v_i and v_j, where the connectivity strength of a path is defined as the product of the weights along that path.

$$d_{ij}^W = \max_{p(i,j)\in\Gamma(i,j)}\left[\prod_{h,l\in p(i,j)} w_{hl}\right] \tag{12}$$

In this measure, the link weights are assumed to be between 0 and 1 to ensure that as the number of links in a path increases, the strength of the path decreases. To measure the directed path length from node v_i to node v_j in a directed binary/weighted network, it is enough that in the above measures the symbol $p(i, j)$ is considered to be a directed path from v_i towards v_j and $\Gamma(i, j)$ a set of all such paths [69,71]. In this case, the directed path length from v_i to v_j is not necessarily equal to the directed path length from v_j to v_i.

2.4. Graph sampling

Existing sampling algorithms can be categorized into two groups as random and topology-based sampling, based on whether nodes or edges are randomly selected from the original graph G (node and edge sampling) or if the selection of nodes and edges depends on the existing topology of G (topology-based sampling).

Classic node sampling (NS) chooses nodes independently and uniformly at random from the graph G. That is, for a required fraction f of nodes, each node is chosen independently with a probability of f for inclusion in the sampled subgraph G_s. Finally, the nodes sampled in V_s along with all the edges among added to E_s constitute the subgraph G_s. While NS is intuitive and relatively straightforward, authors in [119] showed that it does not accurately capture the properties of graphs with power-law degree distribution. Similarly, work in [120] indicated that although classic node sampling captures nodes of different degrees well, due to its inclusion of all the edges for a sampled node-set only, the original level of connectivity is less likely to be preserved. Several variations of node sampling have been proposed in the literature [37,40,47,121,122]. For example, authors in [47] studied the variations of NS in which the selection probability of a node is proportional to either its degree or PageRank weight.

Classic edge sampling (ES) chooses edges independently and uniformly at random from the graph G for inclusion in the sampled subgraph G_s. For each chosen edge, both incident nodes are added to V_s. Finally, G_s is constructed by including the edges sampled in E_s and their endpoints in V_s. Classic edge sampling is likely to capture path lengths, due to its bias towards high degree nodes and the inclusion of both incident nodes of sampled edges [40]. However, ES is less likely to preserve clustering and connectivity, since it samples edges independently [120]. There exist some improved variations of ES in the literature [38,40,47,121,122].

Due to the known limitations of node and edge sampling [47, 119,120], researchers have also considered other sampling methods based on the topological structure of the graph. The common idea in this class of sampling methods is to select a set of initial nodes and then explore the neighborhood of this node-set. The sampled subgraph G_s in these methods is constructed by including the nodes and edges explored. Examples are Breath-First Search (BFS) [123–125], Snowball Sampling (SBS) [126], Forest Fire Sampling (FFS) [127,128], Random Walk Sampling (RWS) [129,130], Respondent-Driven Sampling (RDS) [131,132], and Expansion Sampling [133]. Besides, different variations of Random Walk Sampling have been developed in the literature such as Weighted Random Walk (WRW) [134], Re-Weighted Random Walk (RWRW) [135], Metropolis–Hastings Random Walk (MHRW) [44,136], m-dependent Random Walk (Frontier Sampling) [43], Random Walks with jumps [137], and Random Walk-based Node Sampling (ICLA-NS) [37].

Authors in [138] showed that Random Walk sampling performs much better than NS and ES on the Twitter dataset. Kurant et al. [139] also compared RWS with the BFS method. They found that the average node degree of the original graph is overestimated by BFS since it is biased towards high degree nodes. In contrast, the RW sampling underestimates the average degree. The authors proposed analytical solutions to correct the bias of BFS sampling. Forest Fire sampling behaves almost the same as the BFS method, the difference is that in FFS sampling only a fraction of neighboring nodes is followed at each round. Work in [47] showed that FFS matches very accurately the properties of the original graph. Lee et al. [120] investigated the Snowball sampling method and observed that this method accurately maintains connectivity within the snowball, however, SBS suffers from boundary bias in which the nodes sampled on the last round will be missing a large number of their neighbors. Based on the basic snowball sampling, Gao et al. [39] developed a random multiple snowball with Cohen process sampling. Their simulations indicated that this method can preserve local and global structures of the original graph. Random jump in MHRW sampling with the prevention of being trapped in local structures has been proposed in [42]. Authors in [41] altered the behavior of MHRW sampling by using spirals as a probability distribution instead of a classic normal distribution and showed that their proposed algorithm outperforms normal MHRW in the case of illusion spiral. Ribeiro and Towsley [43] proposed a new approach to an m-dimensional random walk that benefits from starting its walkers at uniformly selected nodes and performs m dependent random walks in the original graph. They showed that their method called Frontier Sampling (FS) mitigates the large estimation errors caused by disconnected or loosely connected graphs which can trap a random walker and distort the estimated characteristics.

Many real-world networks are scale-free, i.e. their degree distribution follows power-law, where $P(k) \sim k^{-\gamma}$, and most of them have $2 < \gamma < 3$ [140]. Authors in [141] showed that these networks are extremely vulnerable to removing hubs (i.e. high degree nodes), breaking into many isolated fragments. Therefore, Yoon et al. [129] proposed that using the weighted sampling, where each node has a weight proportional to its degree, for scale-free networks with $\gamma \lesssim 3$ can produce more connected subgraphs that capture the topological properties of the original graph.

The problem with existing sampling methods is that, in the case of weighted graphs, they ignore edge weights through their sampling process. Therefore, these methods are unable to produce sampled subgraphs that can also match the non-structural properties of weighted graphs. Recently, Rezvanian and Meybodi have considered the weight of edges in the graph sampling process and proposed sampling algorithms for weighted [46] and stochastic [3] networks. In [46], the authors presented the weighted version of some sampling algorithms and tested the performance of their proposed algorithms on various real weighted network benchmarks. They showed that the weighted sampling algorithms can better capture the weighted graph statistics in comparison with their unweighted versions. In [3], the authors addressed the temporal variation of edge weights in social networks and proposed some algorithms for sampling from stochastic graphs where weights associated with edges are random variables. Their experimental results indicated the efficiency of the proposed sampling algorithms, especially eDLAS-SG, in preserving the properties of stochastic networks.

2.5. Learning automata

Stochastic learning automata (LA) [142,143] is considered as an abstract model for adaptive decision making in unknown random environments. LA attempts to learn the optimal action through its repetitive interactions with the unknown environment and in this way improves its performance. At each time step t, the automaton chooses an action $\alpha(t)$ from a finite set of its

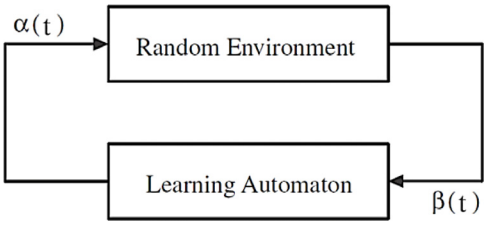

Fig. 2. The interaction between the learning automaton and its random environment.

allowable actions based on the probability distribution kept over the action set, and applies it to the environment. The random environment evaluates the selected action $\alpha(t)$ and responses with a reinforcement signal $\beta(t)$ (either a reward or a penalty) with a certain probability. At last, the automaton updates its action probability distribution depending on the reinforcement signal $\beta(t)$ received from the environment. By these repeated interactions, the learning automaton converges to the optimal action which is the action with the minimum penalty probability. Fig. 2 depicts the interaction between the automaton and its random environment. Learning automata has been widely utilized in literature for solving various problems, such as graph sampling [37,38,144], fuzzy membership function optimization [145,146], trust propagation [28,147], link prediction [148], community detection [149], influence maximization [150].

Stochastic learning automata can be categorized into two main families: variable structure learning automata (VSLA), in which the transition and output functions vary in time, and otherwise, fixed structure learning automata (FSLA). VSLA is represented by a quadruple $\langle \alpha, \beta, p, T \rangle$, where:

- $\alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_r\}$ presents the set of actions from which the automaton chooses.
- $\beta = \{\beta_1, \beta_2, \ldots, \beta_k\}$ presents the set of inputs to the automaton.
- $p = \{p_1, p_2, \ldots, p_r\}$ is the action probability vector with p_i indicating the probability of choosing action α_i.
- T is the learning algorithm used for updating the action probability vector of the automaton depending on the environment's response, i.e. $p(t+1) = T[\alpha(t), \beta(t), p(t)]$, where inputs to the algorithm T are the chosen action $\alpha(t)$, the response $\beta(t)$ of the environment and the action probability vector $p(t)$ at time step t.

Let α_i be the chosen action by the automaton at time step t. The action probability vector $p(t)$ of the automaton is updated according to Eq. (13), if $\alpha_i(t)$ is rewarded by the environment, and it is updated as given in Eq. (14), if $\alpha_i(t)$ is penalized.

$$p_j(t+1) = \begin{cases} p_j(t) + a\left[1 - p_j(t)\right] & j = i \\ (1-a)\, p_j(t) & \forall j \neq i \end{cases} \tag{13}$$

$$p_j(t+1) = \begin{cases} (1-b)\, p_j(t) & j = i \\ \left(\frac{b}{r-1}\right) + (1-b)\, p_j(t) & \forall j \neq i \end{cases} \tag{14}$$

where r is the number of available actions that can be taken by the automaton, a, and b denote reward and penalty parameters which determine the number of increases and decreases in the action probabilities, respectively. If $a = b$, the learning algorithm T is called linear reward–penalty (L_{R-P}) algorithm, if $a \gg b$, T is called linear reward – ϵ penalty ($L_{R-\epsilon P}$) algorithm, and if $b = 0$, it is called linear reward-Inaction (L_{R-I}) algorithm in which the action probability distribution remains unchanged when the chosen action is penalized by the environment. The parameters a and b influence the speed and accuracy of convergence [142]. For a too small value of a, the learning process is too slow and for a too large value of a, the increments in the automaton's action probabilities are too high which results in the low accuracy in learning the optimal action. The probability of convergence to the optimal action can be made as close to 1 as desired by setting the parameter a to a sufficiently small value [142]. In the $L_{R-\epsilon P}$ algorithm, the parameter b is set to a small value in comparison with a ($b = \epsilon a$, where $0 < \epsilon \ll 1$) and the action probability distribution $p(t)$ converges in distribution to a random variable p^*. By choosing ϵ sufficiently small, p^* can be made as close to the optimal probability vector as desired [151].

3. Stochastic trust network

A dynamic graph is one in which any of four entities: the node set, the edge set, the edge weights or the node weights, and or any combination of them, changes over time [152]. In the case of time-varying edge weights, the graph is called an edge-weighted dynamic graph. It is also referred to as a stochastic graph in the literature [153–155], in which the weights assigned to the edges of the graph are random variables with unknown distributions. Considering the dynamic nature of trust, in this paper we propose that using stochastic graphs for modeling trust networks provides a better representation of the properties of these networks. Based on this, the stochastic trust network and sampling from this network are defined as follows.

Definition 1 (*Stochastic Trust Network*)**.** A trust network can be modeled by a stochastic digraph $G(V, E, W)$, where $V = \{v_1, v_2, \ldots, v_N\}$ denotes the set of N nodes (users), $E \subseteq V \times V = \{e_{ij} | v_i, v_j \in V\}$ is the set of M directed links (trust relationships), and $W = \{w_{ij} | e_{ij} \in E\}$ denotes the set of trust weights such that w_{ij} refers to the trust weight corresponding to the trust relationship e_{ij} from v_i to v_j. Since G is a stochastic graph, each trust weight w_{ij} is assumed to be a random variable with an unknown distribution. The trusted neighborhood of the user v_i is defined as $Nei(v_i) = \left\{v_j | e_{ij} \in E, v_j \in V\right\}$, such that $|Nei(v_i)| = d_{out}(v_i)$, the out-degree of a node v_i.

Definition 2 (*Stochastic Trust Network Sampling*)**.** Given a stochastic trust network $G(V, E, W)$ and a sampling fraction f, a dynamic sampling algorithm aims to produce a smaller subgraph $G_s(V_s, E_s, W_s)$ of size $|V_s| = fN$ such that the sampled subgraph G_s preserves the trust-specific statistics of the original graph G.

4. Trust network measures

In this section, we determine the most appropriate existing measures with respect to the special properties of trust and propose new extensions of two graph measures: clustering coefficient and path length, which in addition to considering the topological structure, the directionality, and link weights, also address the properties of trust.

4.1. Degree

To compute degree measure in trust networks, we propose to use the metrics given in [77]. The reason is that Eq. (2) can be considered as a measure of a user v_i's reputation because it is equal to the average of trust statements incoming in v_i, i.e. the average of what all the trusting users of the user v_i think about him, and Eq. (3) can account for the average trust of v_i in its trusted users, both considering the number of users. Since these measures indicate well the importance and position of a user in a trust network, we utilize these definitions for computing the user strength in trust networks.

4.2. Clustering coefficient

In this paper, we propose to consider only transitive triplets for computing the local clustering coefficient in trust networks. The one reason is that an important property of the trust is propagation and transitivity implies propagation. The other reason is that, as shown by Granovetter [156], in social networks strong links are more probable to be part of transitive triplets comparing to weak ones. Therefore, by considering transitive triplets we count the important patterns constituted by high trusted relations around a user when computing the local clustering coefficient. The proposed generalization of this coefficient for a node v_i in a trust network is given as

$$C_i^{DW} = \frac{\sum_{j\neq i}\sum_{h\neq i,j}\left(1-\left|w_{jh}-w'_{jh}\right|\right)a_{ji}a_{ih}a_{jh}}{k_i^{in}k_i^{out}-\sum_{j\neq i}a_{ij}a_{ji}} \tag{15}$$

where the denominator is the total number of possible transitive triplets that v_i can form, as given in Eq. (8). In the numerator, $\left(1-\left|w_{jh}-w'_{jh}\right|\right)$ determines the contribution of each transitive triplet centered on the node v_i to the final result, where $\left|w_{jh}-w'_{jh}\right|$ is the absolute error between the real trust value w_{jh} from v_j to v_h and the propagated trust value w'_{jh} along the path (or chain) from v_j towards v_h. We assume that the trust values are normalized by the maximum trust in the network. The less the absolute error, the more the contribution of the triplet is. There exist two widely used strategies for propagating trust along a path including *Min* and *Multi* [26,35]. Based on these propagation strategies, two following versions of the above equation can be presented:

$$C_i^{DW-min} = \frac{\sum_{j\neq i}\sum_{h\neq i,j}\left(1-\left|w_{jh}-\min(w_{ji},w_{ih})\right|\right)a_{ji}a_{ih}a_{jh}}{k_i^{in}k_i^{out}-\sum_{j\neq i}a_{ij}a_{ji}} \tag{16}$$

$$C_i^{DW-multi} = \frac{\sum_{j\neq i}\sum_{h\neq i,j}\left(1-\left|w_{jh}-w_{ji}w_{ih}\right|\right)a_{ji}a_{ih}a_{jh}}{k_i^{in}k_i^{out}-\sum_{j\neq i}a_{ij}a_{ji}} \tag{17}$$

These equations produce the same result as Eq. (7) when they are applied to a directed binary network and the same result as the standard local clustering coefficient (Eq. (4)) when they are applied to an undirected binary network. Also, it is ensured that $0 \leq C_i^{DW} \leq 1$, because neither numerator nor denominator in any of the fractions can be negative, and the contribution of each existing transitive triplet is at most 1.

4.3. Path length

In trust networks, the strength of a trust path is computed by using a propagation function. As mentioned in Section 4.2, two widely used strategies for propagating trust along a path are *Min* and *Multi*. When multiple trust paths are discovered between two nodes, the trust path with the highest strength is considered the best one [32]. Besides, since trust is diluted through the propagation process, shorter trust paths are favored over longer ones [157]. Based on this, in this paper, we also take into account both path factors, i.e. strength and length, for computing the trust path length from a node v_i to another node v_j and define the following measures considering both propagation functions.

$$d_{ij}^{DW-min} = \min_{p(i,j)\in\Gamma(i,j)}\left[\frac{\sum_{h,l\in p(i,j)}a_{hl}}{\min_{h,l\in p(i,j)}w_{hl}}\right] \tag{18}$$

$$d_{ij}^{DW-multi} = \min_{p(i,j)\in\Gamma(i,j)}\left[\frac{\sum_{h,l\in p(i,j)}a_{hl}}{\prod_{h,l\in p(i,j)}w_{hl}}\right] \tag{19}$$

where for each path $p(i,j)$ from v_i towards v_j, the denominator in Eqs. (18) and (19) computes the strength of the path respectively in terms of Min and Multi propagation functions, and the numerator denotes the length of the path in both equations. When links are unweighted, these definitions produce the same results as Eq. (9).

5. The proposed dynamic sampling algorithm

In this section, we propose a learning automata-based dynamic sampling algorithm for stochastic trust networks, called D-LAFS, and describe its process in detail.

The proposed algorithm D-LAFS is an extension of Frontier sampling (FS): an approach to the m-dimensional random walk which starts from a collection of m randomly selected nodes and performs m dependent random walks in the original graph. By using learning automata, D-LAFS attempts to guarantee the connectivity of subgraphs sampled by FS and, at the same time, capture the changes of edge weights through the sampling process in such a way that the number of samples taken from the edges is reduced as much as possible. The pseudo-code for the algorithm D-LAFS is given in Fig. 3. The inputs to D-LAFS are the stochastic trust network $G(V,E,W)$, the sampling fraction f, the size m of seed set and the stopping threshold $\mathfrak{T}$, and the output is a smaller subgraph $G_s(V_s,E_s,W_s)$ of size $|V_s| = fN$ which preserves accurately the trust-specific statistics of G. At first, a network of learning automata is formed by equipping each node $v_i \in V$ with an automaton A_i whose action set α_i includes the trusted neighborhood of v_i, i.e. $\alpha_i = Nei(v_i)$. The action probability vector of each learning automaton A_i is initialized as given in Eq. (20). Since there might be more than one optimal subgraph for the original graph, we use the learning algorithm $L_{R-\epsilon P}$ for updating the action probability distributions of automata (for more details, see Section 2.5).

$$p_i^j(0) = \frac{w_{ij}}{\sum_{v_k\in Nei(v_i)}w_{ik}} \qquad v_j \in Nei(v_i) \tag{20}$$

where $p_i^j(0)$ denotes the probability that the trusted neighbor v_j is chosen by A_i at stage $t=0$. Using this equation for initializing the probability vectors of learning automata, highly trusted neighbors are initially given higher probabilities of being selected.

After constructing a network of the learning automata isomorphic to the trust network G, D-LAFS randomly selects a set of m nodes of G as the starting points for random walkers, called seed set $\mathfrak{S}$. The algorithm then iteratively constructs a sample subgraph and updates the action probability vectors of the learning automata until the stop condition is reached. Stage t of the proposed algorithm D-LAFS is described in the following three steps.

Step 1. Constructing a sample subgraph

This step aims to construct a sample subgraph from the original graph G by m chains of automaton activations starting from the seed set $\mathfrak{S}$ as follows. At first, one of the nodes in $\mathfrak{S}$, say v_i, is randomly selected with probability $d_{out}(v_i)/\sum_{v_j\in\mathfrak{S}}d_{out}(v_j)$. If the corresponding automaton A_i has been already activated in the current stage or there is no available action for it (i.e. $d_{out}(v_i)=0$), the current chain cannot continue forward and thus another seed node will be selected. Otherwise, A_i is activated to determine the next hop along its chain. The activated automaton A_i chooses one of its actions, say v_j, based on the action probability distribution. The seed set $\mathfrak{S}$ is updated by replacing the node v_i by v_j. A sample is taken from the edge weight w_{ij} and the edge e_{ij} along with its incident nodes v_i and v_j are added to the subgraph G_s. The process of selecting a seed node and activating its corresponding automaton is repeated until the required number of nodes is sampled (i.e. $|V_s| = fN$) or there is no chain to be continued.

Step 2. Updating the action probability vectors

Algorithm D-LAFS : LA-based dynamic sampling

Input Original trust network $G(V, E, W)$, Fraction of nodes f, Size of seed set m, Threshold $\mathcal{T}$
Output Sampled subgraph $G_s(V_s, E_s, W_s)$
Begin
  Let t be the stage number and is initially set to 0
  Associate each node $v_i \in G$ with an automaton A_i with $d_{out}(v_i)$ actions
  Initialize the action probability vector of each A_i according to Eq. 3
  Let $\mathcal{S}$ denotes the seed set and initially includes m nodes randomly selected from G
  Repeat
    Let $\mathcal{C}$ be the set of traversed chains at stage t with $\mathcal{C}_i$ indicating ith chain and is initially set to $\mathcal{S}$
    Let $\mathcal{L}$ be the set of ending nodes of traversable chains in $\mathcal{C}$ at stage t and is initially set to $\mathcal{S}$
    $E_s \leftarrow \{\ \}$; $V_s \leftarrow \{\ \}$
    Repeat
      Select a node $v_i \in \mathcal{L}$ with a probability proportional to its degree
      If $(d_{out}(v_i) = 0$ **or** A_i has been already activated at stage $t)$ **Then**
        $\mathcal{L} \leftarrow \mathcal{L} \backslash v_i$
      Else
        Automaton A_i is activated and chooses an action v_j according to its action probability vector
        $\mathcal{L} \leftarrow \mathcal{L} \backslash v_i \cup \{v_j\}$
        Add v_j to the end of the chain $\mathcal{C}_i$
        Take a sample from the edge weight w_{ij}
        $E_s \leftarrow E_s \cup \{e_{ij}\}$; $V_s \leftarrow V_s \cup \{v_i, v_j\}$
      End If
    Until $(|V_s| = f|V|$ **or** $\mathcal{L} = \emptyset)$
    For (each chain $\mathcal{C}_i \in \mathcal{C}$) **Do**
      If $(|V_s| = f|V|$ **and** $\mathcal{C}_i$ has at least one node in common with any other chain) **Then**
        Reward all the activated learning automata along $\mathcal{C}_i$
      Else
        Penalty all the activated learning automata along $\mathcal{C}_i$
      End If
    End For
    $t \leftarrow t + 1$
  Until (all the activated LAs are rewarded **or** $t > \mathcal{T}$)
End Algorithm

Fig. 3. The pseudo code of the D-LAFS algorithm.

In this step, the chains forming the sampled subgraph in the previous step are evaluated and the action probability vectors of the learning automata are updated in such a way that the activated learning automata along a chain receive a reward if the chain has at least one node in common with any other chain and the total number of sampled nodes is equal to fN (i.e. $|V_s| = fN$), and otherwise, they are penalized. In this way, learning automata learn to sample connected subgraphs with the required size.

Step 3. Checking stop condition

The algorithm terminates when all the activated learning automata in the current stage are rewarded (or the sample subgraph G_s is connected and $|V_s| = fN$) or the number of stages exceeds a certain threshold $\mathcal{T}$. In this situation, the subgraph G_s is considered as the final sampled subgraph.

6. Experimental evaluation

To investigate the efficiency of our proposed dynamic sampling algorithm D-LAFS, we conduct several extensive experiments on the well-known trust network datasets including Advogato, Squeakfoundation, and Robots_net, and compare results obtained for D-LAFS with those of existing sampling algorithms. In the proposed algorithm D-LAFS, we use the learning algorithm $L_{R-\epsilon P}$ for updating the action probability vectors of learning automata and set the threshold $\mathcal{T}$ to 100. The size m of the seed set is set to 50, 30, and 10 respectively for three test datasets Advogato, Robots_net, and Squeakfoundation. In the experiments, results are reported as an average of 10 independent runs for the sampling fraction f ranging from 0.1 to 0.3 with increment 0.05.

6.1. Experimental design

6.1.1. Methods for comparison

In the experiments, we compare the results obtained for D-LAFS with those of two well-known sampling algorithms, including Random Walk Sampling (RWS) and Frontier Sampling (FS), as well as those of the stochastic sampling algorithm eDLAS-SG presented in [3]. For the comparison purpose, we extend both algorithms RWS and FS in such a way that each time an edge is traversed by the algorithm, a sample will be taken from the edge weight and the average weight of the edge is updated. The dynamic version of these algorithms is described as follows.

Table 1
Characteristics of test datasets.

Dataset	Nodes	Edges	Density
Advogato	5417	51312	0.002
Squeakfoundation	461	2697	0.013
Robots_net	1723	3596	0.001

Dynamic Random Walk Sampling (D-RWS). The algorithm starts from an initial node. Then, at stage t the random walker at node v_i selects an outgoing edge e_{ij} (or a neighboring node $v_j \in Nei(v_i)$) uniformly at random from the set of outgoing edges of v_i. As a result of this edge selection, a sample is taken from the edge weight w_{ij}, and the edge e_{ij} along with its incident nodes v_i and v_j are added to the subgraph G_s. At the next stage, the walker starts from v_j and the sampling process continues until $|V_s| = f|V|$.

Dynamic Frontier Sampling (D-FS). The algorithm starts with a collection $\mathfrak{S}$ of m randomly chosen nodes. Then, at each stage t a node $v_i \in \mathfrak{S}$ is chosen with probability $d_{out}(v_i) / \sum_{v_j \in \mathfrak{S}} d_{out}(v_j)$. The walker at v_i selects an outgoing edge e_{ij} uniformly at random. As a result, the node v_i is replaced by v_j in the set $\mathfrak{S}$, a sample is taken from w_{ij}, and the subgraph G_s is updated by adding e_{ij} along with its incident nodes v_i and v_j. This process is repeated until the required number of nodes is sampled.

6.1.2. Dataset

The experiments are conducted on three well-known trust network datasets: Advogato, Squeakfoundation, and Robots_net borrowed from Trustlet (www.trustlet.org). Advogato is an online social networking website for free software developers which consists of 51,312 trust statements issued by 5417 users. The dataset has extracted from advogato.org on July 07, 2014. Squeakfoundation contains 461 users and 2697 trust statements extracted from squeak.org on November 29, 2008. Robots_net is a web community site related to robotics that has 1723 users who have expressed 3596 trust statements. The dataset has extracted from robots.net on July 07, 2014. Users in these networks are allowed to certify each other on four different trust levels: Observer, Apprentice, Journeyer, and Master. A summary of the global statistics of these three datasets has been reported in Table 1.

To perform the experimental evaluation, we need trust networks to be stochastic graphs in which each trust relationship is associated with a probability distribution. For this purpose, we utilize the technique proposed in [158] which has been widely used in the literature [26,28,36,159]. In this technique, each user v_i in the network has a quality measurement $q_i \in [0, 1]$ that determines the probability that a trust statement issued by the user is true. In this paper, we consider the reputation of a user as its quality value. That is, the quality q_i of a user v_i is computed as the average of the trust statements incoming in the user v_i. Since trust statements in the three datasets must be real-valued in $[0, 1]$, we assign 0.2 to Observer, 0.6 to Apprentice, 0.8 to Journeyer, and 1 to Master. After that, the trust weight w_{ij} for each trust relationship e_{ij} from v_i to v_j is assumed to be a random variable with a continuous uniform distribution on the interval $\left[\max\left(q_j - \delta_{ij}, 0\right), \min(q_j + \delta_{ij}, 1)\right]$, where the noise parameter $\delta_{ij} = \frac{1-q_i}{2}$ determines how accurate user v_i is at estimating the quality of its trusted user v_j.

6.1.3. Evaluation metric

In this paper, we utilize the distance measure Kolmogorov–Smirnov (KS) for the experimental evaluation and briefly describe it below.

Kolmogorov–Smirnov (KS) statistic. The KS statistic assesses the maximum vertical distance between two cumulative distribution functions (CDF) as given in Eq. (21), where x is the range of the random variable, F and F' denotes two CDFs, and $0 \leq KS \leq 1$.

$$KS = \max_x \left|F(x) - F'(x)\right| \tag{21}$$

This statistic has been widely used as a measure of the agreement between two cumulative distributions [37,38,40].

6.2. Experimental results

6.2.1. Experiment I

In this experiment, we examine the impact of the learning parameters a and b on the performance of our proposed algorithm D-LAFS. For this purpose, various settings for these parameters are considered and for each setting, we compute the KS distance averaged on the three datasets for different sampling fractions varying from 0.1 to 0.3. Fig. 4(a-d) show the results respectively for in-degree, out-degree, clustering coefficient, and path length distributions. As one can see from these figures, by setting $a = 0.03, b = 0.0003$ D-LAFS obtains the least KS distance for three statistics in-degree, out-degree, and path length distributions, and for the clustering coefficient statistic, it obtains the results close to the best ones. Therefore, we use this setting in our algorithm D-LAFS for the experimentations that follow.

6.2.2. Experiment II

This experiment aims to study the ability of the proposed algorithm D-LAFS in preserving the statistics of stochastic trust networks comparing to the other sampling algorithms, including Dynamic Random Walk Sampling (D-RWS), Dynamic Frontier Sampling (D-FS), and eDLAS-SG [3]. We consider the statistics in-degree, out-degree, clustering coefficient, and path length distributions and for each statistic, we plot the average KS distance taken over all test datasets for different algorithms in Fig. 5(a-d).

As shown in these figures, our algorithm D-LAFS yields the highest performance for all the graph statistics compared to the other algorithms, and D-RWS performs the worst. For in-degree and path length, D-LAFS performs significantly better than the others and for the out-degree distribution, D-FS and eDLAS-SG provide the results almost close to D-LAFS. Considering these three statistics, D-FS provides better results than eDLAS-SG for the sampling fractions up to 0.15 and performs worse for the sampling fractions larger than 0.15. D-RWS obtains the most KS distance for the in-degree, out-degree, and path length distributions. For the clustering coefficient statistic, D-RWS performs almost as good as D-FS and the algorithm eDLAS-SG provides the results almost close to D-LAFS.

From these figures, we can also see that increasing the sampling fraction decreases the average KS distance. The decrement is much faster for the algorithm eDLAS-SG for three statistics in-degree, out-degree, and path length distributions.

In this experiment, we also report the KS distance of the algorithms for each dataset at the sampling fraction 0.2. The results for in-degree, out-degree, clustering coefficient, and path length distributions are shown in Table 2. Based on this table, D-LAFS significantly outperforms the other algorithms on Advogato and Robots_net for all the graph statistics. On the Squeakfoundation dataset, D-LAFS performs almost as good as eDLAS-SG for clustering coefficient distribution but preserves more accurately the in-degree and path length statistics. For the out-degree distribution, D-LAFS performs the worst on Squeakfoundation. eDLAS-SG provides better results than D-RWS and D-FS on Squeakfoundation for four statistics. on Advogato and Robots_net datasets, D-FS provides better results than D-RWS and eDLAS-SG for in-degree, out-degree, and path length distributions, and for clustering coefficient statistic, eDLAS-SG outperforms two other algorithms.

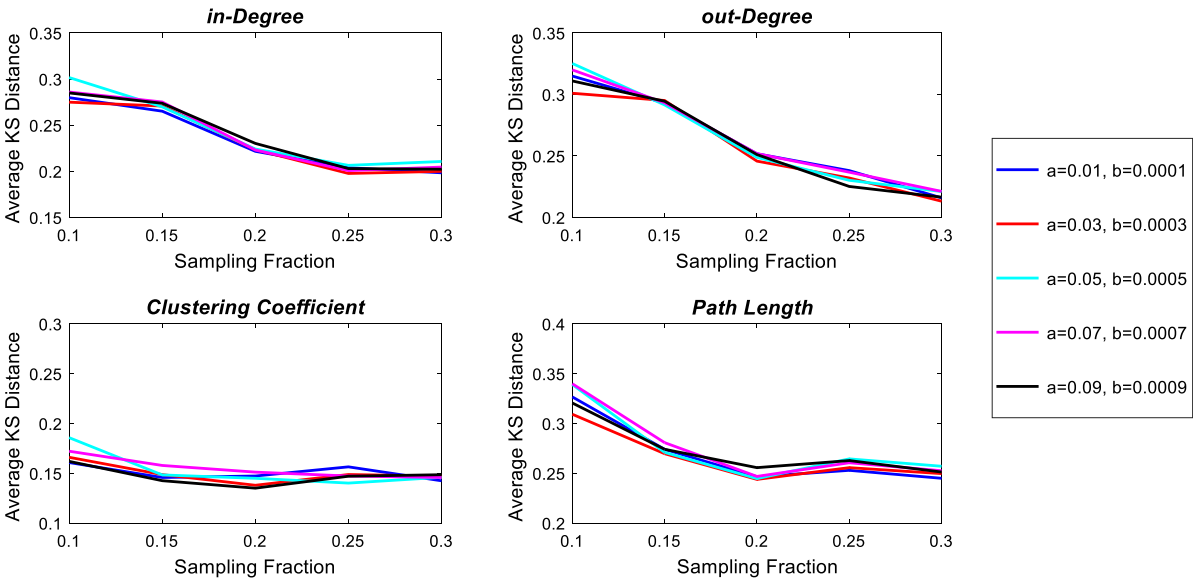

Fig. 4. The impact of learning parameters on the performance of D-LAFS. Generally, D-LAFS obtains the least KS distance by setting $a = 0.03, b = 0.0003$.

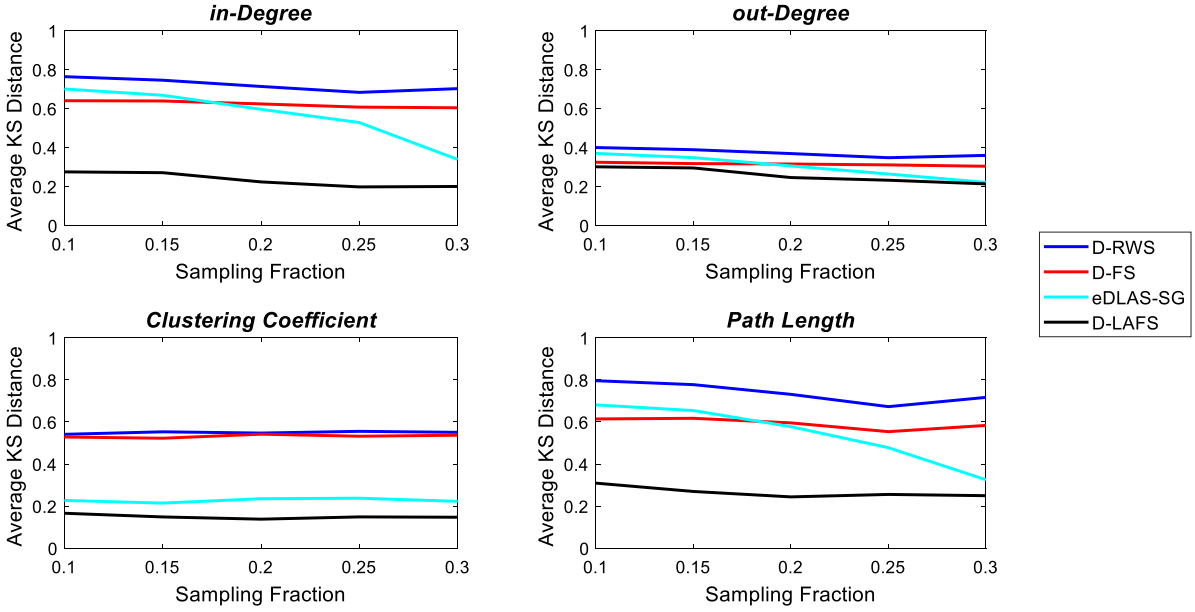

Fig. 5. Comparing different dynamic sampling algorithms. Generally, increasing the sampling fraction decreases the average KS distance. D-LAFS yields the best performance especially for in-degree and path length compared to the other algorithms.

As stated in Table 1, the Advogato and Robots_net datasets have much lower density compared to the Squeakfoundation dataset. Considering the results reported in Table 2, we can say that our proposed algorithm D-LAFS performs better in networks that are generally sparse, which is the case for many real-world networks [160].

6.2.3. Experiment III

In this experiment, we investigate the number of disjoint connected components in subgraphs sampled by our algorithm D-LAFS at the sampling fraction 0.2. The less the number of disjoint components, the more the connectivity is in the sampled subgraph. Table 3 compares the number of connected components for D-LAFS with that of the other algorithms and with the real number of them for each dataset. According to the results of this table, our proposed algorithm D-LAFS produces more connected subgraphs than the other algorithms. For Squeakfoundation and Robots_net, D-RWS includes the most number of disjoint components in its sampled subgraphs and eDLAS-SG provides more connectivity than D-FS. For the Advogato dataset, eDLAS-SG has the highest disconnectivity and D-FS includes a less number of disjoint components compared to D-RWS.

By using learning automata, the algorithm D-LAFS attempts to alleviate the disconnectivity problem of sampled subgraphs by D-FS. As one can see, D-LAFS produces connected subgraphs for the Squeakfoundation dataset which originally includes four disjoint components.

7. Conclusion

Trust is built up over time as a result of repeated direct interactions between users. As two parties frequently interact with each other, their relation strengthens, and trust between them evolves based on their experience of interactions. Trust increases

Table 2
KS distance for all test datasets, at sampling fraction 0.2.

Statistic	Dataset	D-RWS	D-FS	eDLAS-SG	D-LAFS
in-degree	*Advogato*	0.5657	0.5688	0.6370	0.1170
	Squeakfoundation	0.8586	0.7532	0.5387	0.3454
	Robots_net	0.7163	0.5496	0.6134	0.2081
	Avg. for all datasets	0.7135	0.6239	0.5964	**0.2235**
out-degree	*Advogato*	0.5163	0.4703	0.5375	0.0586
	Squeakfoundation	0.2848	0.2526	0.1205	0.5202
	Robots_net	0.3056	0.2229	0.2585	0.1586
	Avg. for all datasets	0.3689	0.3153	0.3055	**0.2458**
Clustering coefficient	*Advogato*	0.5377	0.5399	0.2021	0.0556
	Squeakfoundation	0.7719	0.7628	0.2213	0.2222
	Robots_net	0.3327	0.3236	0.2821	0.1364
	Avg. for all datasets	0.5474	0.5421	0.2352	**0.1380**
Path length	*Advogato*	0.6669	0.5807	0.6479	0.0768
	Squeakfoundation	0.6877	0.5870	0.4391	0.4094
	Robots_net	0.8397	0.6194	0.6464	0.2457
	Avg. for all datasets	0.7314	0.5957	0.5778	**0.2439**

Table 3
The number of connected components in sample subgraphs versus its real value for each dataset, at sampling fraction 0.2.

Dataset	Real connected components	D-RWS	D-FS	eDLAS-SG	D-LAFS
Advogato	69	8.6	6.1	25.6	**3.1**
Squeakfoundation	4	5.9	3.1	1.1	**1**
Robots_net	106	15.2	10.6	7.8	**3.8**

if the experience is positive and otherwise it decreases. Since trust is dynamic and time-variable, a stochastic graph in which edge weights are random variables is a more appropriate structure to model the dynamicity property of the trust in trust networks. However, most works in the scope of trust have been modeled trust networks as static graphs. In this paper, we proposed using a stochastic graph model for representing trust networks and introduced the concept of the stochastic trust network.

Trust networks are of the most important social networks. The study of these networks requires sampling methods that not only capture the trust weights and their variations over time through the sampling process but also be able to preserve properties of trust networks. To the best of our knowledge, this paper is the first to address this issue. Since there exist various generalizations of graph measures, which take into consideration topological structure, directionality, and weights of edges, we reviewed the literature for most appropriate generalizations with respect to the properties of trust and proposed new generalizations wherever existing ones were not sufficient, in particular for the clustering coefficient and path length measures. Considering the trust-specific measures, we proposed a dynamic algorithm based on learning automata for sampling from stochastic trust networks which is an extension of the Frontier sampling (FS) algorithm. By using learning automata, the proposed algorithm called D-LAFS, in addition to answering the above needs, can also tackle the disconnectivity problem of sampled subgraphs by FS. We investigated the performance of D-LAFS by conducting several experiments on the well-known trust network datasets. Our experimental results showed that the proposed algorithm D-LAFS outperforms the existing sampling methods and preserves more accurately the trust-specific measures of trust networks.

CRediT authorship contribution statement

Mina Ghavipour: Conceptualization, Methodology, Formal analysis, Investigation, Software, Visualization, Validation, Writing - original draft, Writing - review & editing, Funding acquisition. **Mohammad Reza Meybodi:** Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This research was in part supported by a grant from the Institute for Research in Fundamental Sciences (IPM), Iran.

References

[1] A. Rezvanian, M.R. Meybodi, Stochastic graph as a model for social networks, Comput. Human Behav. 64 (2016) 621–640.
[2] B. Moradabadi, M.R. Meybodi, Link prediction in stochastic social networks: Learning automata approach, J. Comput. Sci. (2017) https://doi.org/10.1016/j.jocs.2017.08.007.
[3] A. Rezvanian, M.R. Meybodi, Sampling algorithms for stochastic graphs: A learning automata approach, Knowl.-Based Syst. 127 (2017) 126–144.
[4] M. Ghavipour, M.R. Meybodi, A dynamic algorithm for stochastic trust propagation in online social networks: Learning automata approach, Comput. Commun. 123 (2018) 11–23, https://doi.org/10.1016/j.comcom.2018.04.004.
[5] M. Ghavipour, M.R. Meybodi, Stochastic trust network enriched by similarity relations to enhance trust-aware recommendations, Appl. Intell. (2018) 1–14.
[6] L. Xiong, L. Liu, A reputation-based trust model for peer-to-peer e-commerce communities, in: E-Commerce, 2003. CEC 2003. IEEE Int. Conf, IEEE, 2003, pp. 275–284.
[7] L. Xiong, L. Liu, Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities, IEEE Trans. Knowl. Data Eng. 16 (2004) 843–857.
[8] J. Zhang, R. Cohen, Evaluating the trustworthiness of advice about seller agents in e-marketplaces: A personalized approach, Electron. Commer. Res. Appl. 7 (2008) 330–340.
[9] R. Chen, F. Bao, M. Chang, J.-H. Cho, Dynamic trust management for delay tolerant networks and its application to secure routing, IEEE Trans. Parallel Distrib. Syst. 25 (2014) 1200–1210, https://doi.org/10.1109/TPDS.2013.116.
[10] X. Liu, A. Datta, A trust prediction approach capturing agents' dynamic behavior, in: IJCAI Proceedings-International Jt. Conf. Artif. Intell, Citeseer, 2011, p. 2147, https://doi.org/10.5591/978-1-57735-516-8/IJCAI11-358.
[11] R. Wishart, R. Robinson, J. Indulska, A. Jøsang, Superstringrep: reputation-enhanced service discovery, in: Proc. Twenty-Eighth Australas. Conf. Comput. Sci. 38, Australian Computer Society, Inc., 2005, pp. 49–57.
[12] S.D. Kamvar, M.T. Schlosser, H. Garcia-Molina, The eigentrust algorithm for reputation management in p2p networks, in: Proc. 12th Int. Conf. World Wide Web, ACM, 2003, pp. 640–651.
[13] G. von Laszewski, B.K. Alunkal, I. Veljkovic, Toward reputable grids, Scalable Comput. Pract. Exp. 6 (2001).
[14] S. Song, K. Hwang, R. Zhou, Y.-K. Kwok, Trusted P2p transactions with fuzzy reputation aggregation, IEEE Internet Comput. 9 (2005) 24–34.

[15] Y. Zhang, Y. Fang, A fine-grained reputation system for reliable service selection in peer-to-peer networks, IEEE Trans. Parallel Distrib. Syst. 18 (2007) https://doi.org/10.1109/TPDS.2007.1043.
[16] C.J. Fung, J. Zhang, I. Aib, R. Boutaba, Dirichlet-Based trust management for effective collaborative intrusion detection networks, IEEE Trans. Netw. Serv. Manag. 8 (2011) 79–91.
[17] R. Zhou, K. Hwang, Powertrust: A robust and scalable reputation system for trusted peer-to-peer computing, IEEE Trans. Parallel Distrib. Syst. 18 (2007) https://doi.org/10.1109/TPDS.2007.1021.
[18] Z. Su, M. Li, X. Fan, X. Jin, Z. Wang, Research on trust propagation models in reputation management systems, Math. Probl. Eng. 2014 (2014) https://doi.org/10.1155/2014/536717.
[19] R. Jurca, B. Faltings, An incentive compatible reputation mechanism, in: E-Commerce, 2003. CEC 2003. IEEE Int. Conf, IEEE, 2003, pp. 285–292.
[20] B. Yu, M.P. Singh, Distributed reputation management for electronic commerce, Comput. Intell. 18 (2002) 535–549.
[21] O. Ayadi, N. Halouani, F. Masmoudi, A fuzzy collaborative assessment methodology for partner trust evaluation, Int. J. Intell. Syst. 31 (2016) 488–501.
[22] K.K. Bharadwaj, M.Y.H. Al-Shamri, Fuzzy computational models for trust and reputation systems, Electron. Commer. Res. Appl. 8 (2009) 37–47, https://doi.org/10.1016/j.elerap.2008.08.001.
[23] G. Wang, J. Wu, Multi-dimensional evidence-based trust management with multi-trusted paths, Futur. Gener. Comput. Syst. 27 (2011) 529–538, https://doi.org/10.1016/j.future.2010.04.015.
[24] F.J. Ortega, J.A. Troyano, F.L. Cruz, C.G. Vallejo, F. Enríquez, Propagation of trust and distrust for the detection of trolls in a social network, Comput. Networks. 56 (2012) 2884–2895, https://doi.org/10.1016/j.comnet.2012.05.002.
[25] Y.A. Kim, H.S. Song, Strategies for predicting local trust based on trust propagation in social networks, Knowl.-Based Syst. 24 (2011) 1360–1371, https://doi.org/10.1016/j.knosys.2011.06.009.
[26] W. Jiang, J. Wu, F. Li, G. Wang, H. Zheng, Trust evaluation in online social networks using generalized network flow, IEEE Trans. Comput. 65 (2016) 952–963.
[27] U. Kuter, J. Golbeck, Using probabilistic confidence models for trust inference in web-based social networks, ACM Trans. Internet Technol. 10 (2010) 8, https://doi.org/10.1145/1754393.1754397.
[28] M. Ghavipour, M.R. Meybodi, Trust propagation algorithm based on learning automata for inferring local trust in online social networks, Knowl.-Based Syst. (2017) https://doi.org/10.1016/j.knosys.2017.06.034.
[29] V. Kant, K.K. Bharadwaj, Fuzzy computational models of trust and distrust for enhanced recommendations, Int. J. Intell. Syst. 28 (2013) 332–365.
[30] M. Lesani, N. Montazeri, Fuzzy trust aggregation and personalized trust inference in virtual social networks, Comput. Intell. 25 (2009) 51–83, https://doi.org/10.1111/j.1467-8640.2009.00334.x.
[31] F. Hao, G. Min, M. Lin, C. Luo, L.T. Yang, Mobifuzzytrust: an efficient fuzzy trust inference mechanism in mobile social networks, IEEE Trans. Parallel Distrib. Syst. 25 (2014) 2944–2955.
[32] Y.A. Kim, An enhanced trust propagation approach with expertise and homophily-based trust networks, Knowl.-Based Syst. 82 (2015) 20–28, https://doi.org/10.1016/j.knosys.2015.02.023.
[33] S. Lyu, J. Liu, M. Tang, Y. Xu, J. Chen, Efficiently predicting trustworthiness of mobile services based on trust propagation in social networks, Mob. Networks Appl. 20 (2015) 840–852, https://doi.org/10.1007/s11036-015-0619-y.
[34] G.F. Liu, Y. Wang, M.A. Orgun, E.-P.P. Lim, Finding the optimal social trust path for the selection of trustworthy service providers in complex social networks, Ieee Trans. Serv. Comput. 6 (2013) 152–167, https://doi.org/10.1109/Tsc.2011.58.
[35] W. Jiang, G. Wang, J. Wu, Generating trusted graphs for trust evaluation in online social networks, Futur. Gener. Comput. Syst. 31 (2014) 48–58.
[36] S. Shekarpour, S.D. Katebi, Modeling and evaluation of trust with an extension in semantic web, Web Semant. Sci. Serv. Agents World Wide Web. 8 (2010) 26–36.
[37] M. Ghavipour, M.R. Meybodi, Irregular cellular learning automata-based algorithm for sampling social networks, Eng. Appl. Artif. Intell. 59 (2017) 244–259, https://doi.org/10.1016/j.engappai.2017.01.004.
[38] M. Ghavipour, M.R. Meybodi, A streaming sampling algorithm for social activity networks using fixed structure learning automata, Appl. Intell. (2017) https://doi.org/10.1007/s10489-017-1005-1.
[39] Q. Gao, X. Ding, F. Pan, W. Li, An improved sampling method of complex network, Int. J. Mod. Phys. C. 25 (2014) 1440007.
[40] N.K. Ahmed, J. Neville, R. Kompella, Network sampling: From static to streaming graphs, ACM Trans. Knowl. Discov. Data. 8 (2014) 7, https://doi.org/10.1145/2601438.
[41] C.A. Piña García, D. Gu, Spiraling facebook: an alternative Metropolis–Hastings random walk using a spiral proposal distribution, Soc. Netw. Anal. Min. 3 (2013) 1403–1415.
[42] L. Jin, Y. Chen, P. Hui, C. Ding, T. Wang, A.V. Vasilakos, B. Deng, X. Li, Albatross sampling: robust and effective hybrid vertex sampling for social graphs, in: Proc. 3rd ACM Int. Work. MobiArch, ACM, 2011, pp. 11–16.
[43] B. Ribeiro, D. Towsley, Estimating and sampling graphs with multidimensional random walks, in: Proc. 10th ACM SIGCOMM Conf. Internet Meas, ACM, 2010, pp. 390–403.
[44] C.-H. Lee, X. Xu, D.Y. Eun, C.-H. Lee, X. Xu, D.Y. Eun, Beyond random walk and metropolis-hastings samplers, in: Proc. 12th ACM SIGMETRICS/PERFORMANCE Jt. Int. Conf. Meas. Model. Comput. Syst. - SIGMETRICS '12, ACM Press, New York, New York, USA, 2012, p. 319, https://doi.org/10.1145/2254756.2254795.
[45] A. Barrat, M. Barthelemy, R. Pastor-Satorras, A. Vespignani, The architecture of complex weighted networks, Proc. Natl. Acad. Sci. USA 101 (2004) 3747–3752.
[46] A. Rezvanian, M.R. Meybodi, Sampling algorithms for weighted networks, Soc. Netw. Anal. Min. 6 (2016) 60.
[47] J. Leskovec, C. Faloutsos, Sampling from large graphs, in: In: Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. - KDD '06, ACM Press, New York, New York, USA, 2006, p. 631, https://doi.org/10.1145/1150402.1150479.
[48] U. Kuter, J. Golbeck, Sunny: A new algorithm for trust inference in social networks using probabilistic confidence models, in: AAAI, 2007, pp. 1377–1382.
[49] W. Sherchan, S. Nepal, C. Paris, A survey of trust in social networks, ACM Comput. Surv. 45 (2013) 47, https://doi.org/10.1145/2501654.2501661.
[50] B. Yu, M.P. Singh, K. Sycara, Developing trust in large-scale peer-to-peer systems, in: IEEE First Symp. OnMulti-Agent Secur. Surviv. 2004, IEEE, 2004, pp. 1–10.
[51] L. Mui, M. Mohtashemi, A. Halberstadt, A computational model of trust and reputation, in: Syst. Sci. 2002. HICSS. Proc. 35th Annu. Hawaii Int. Conf, IEEE, 2002, pp. 2431–2439.
[52] J. Sabater i Mir, Trust and Reputation for Agent Societies, Universitat Autònoma de Barcelona, 2004.
[53] M. Schillo, P. Funk, M. Rovatsos, Using trust for detecting deceitful agents in artificial societies, Appl. Artif. Intell. 14 (2000) 825–848.
[54] B. Christianson, W.S. Harbison, Why isn't trust transitive? in: Int. Work. Secur. Protoc, Springer, 1996, pp. 171–176.
[55] B. Yu, M.P. Singh, A social mechanism of reputation management in electronic communities, in: Int. Work. Coop. Inf. Agents, Springer, 2000, pp. 154–165.
[56] I. Yaniv, E. Kleinberger, Advice taking in decision making: Egocentric discounting and reputation formation, Organ. Behav. Hum. Decis. Process. 83 (2000) 260–281.
[57] S. Staab, B. Bhargava, L. Leszek, A. Rosenthal, M. Winslett, M. Sloman, T.S. Dillon, E. Chang, F. Hussain, W. Nejdl, The pudding of trust: Managing the dynamic nature of trust, IEEE Intell. Syst. 19 (2004) 74–88.
[58] J. Guo, R. Chen, J.J.P. Tsai, A survey of trust computation models for service management in internet of things systems, Comput. Commun. 97 (2017) 1–14.
[59] L. Gyarmati, T.A. Trinh, Measuring user behavior in online social networks, IEEE Netw. 24 (2010).
[60] A.J. Morales, J.C. Losada, R.M. Benito, Users structure and behavior on an online social network during a political protest, Phys. A Stat. Mech. Appl. 391 (2012) 5244–5253.
[61] Q. Yan, L. Wu, L. Zheng, Social network based microblog user behavior analysis, Phys. A Stat. Mech. Appl. 392 (2013) 1712–1723.
[62] D.R. Bild, Y. Liu, R.P. Dick, Z.M. Mao, D.S. Wallach, Aggregate characterization of user behavior in Twitter and analysis of the retweet graph, ACM Trans. Internet Technol. 15 (2015) 4.
[63] S. Guo, M. Wang, J. Leskovec, The role of social networks in online shopping: information passing, price of trust, and consumer choice, in: Proc. 12th ACM Conf. Electron. Commer, ACM, 2011, pp. 157–166.
[64] G. Fagiolo, Clustering in complex directed networks, Phys. Rev. E. 76 (2007) 26107.
[65] S.E. Ahnert, D. Garlaschelli, T.M.A. Fink, G. Caldarelli, Ensemble approach to the analysis of weighted networks, Phys. Rev. E. 76 (2007) 16101.
[66] I.E. Antoniou, E.T. Tsompa, Statistical analysis of weighted networks, Discret. Dyn. Nat. Soc. 2008 (2008).
[67] M.E.J. Newman, Analysis of weighted networks, Phys. Rev. E. 70 (2004) 56131.
[68] J.-P. Onnela, J. Saramäki, J. Hyvönen, G. Szabó, M.A. De Menezes, K. Kaski, A.-L. Barabási, J. Kertész, Analysis of a large-scale weighted network of one-to-one human communication, New J. Phys. 9 (2007) 179.
[69] W. Liao, J. Ding, D. Marinazzo, Q. Xu, Z. Wang, C. Yuan, Z. Zhang, G. Lu, H. Chen, Small-world directed networks in the human brain: multivariate granger causality analysis of resting-state fMRI, Neuroimage. 54 (2011) 2683–2694.
[70] B.M. Tabak, M. Takami, J.M.C. Rocha, D.O. Cajueiro, S.R.S. Souza, Directed clustering coefficient as a measure of systemic risk in complex banking networks, Phys. A Stat. Mech. Appl. 394 (2014) 211–216.

[71] C. Yan, Y. He, Driving and driven architectures of directed small-world human brain functional networks, PLoS One 6 (2011) e23460.
[72] J.-P. Onnela, J. Saramäki, J. Kertész, K. Kaski, Intensity and coherence of motifs in weighted complex networks, Phys. Rev. E. 71 (2005) 65103.
[73] B. Zhang, S. Horvath, A general framework for weighted gene co-expression network analysis, Stat. Appl. Genet. Mol. Biol. 4 (2005) 1128, https://doi.org/10.2202/1544-6115.1128.
[74] M. Kivelä, Weighted and temporal networks: Towards more realistic representations of complex systems, 2012, http://urn.fi/URN:ISBN:978-952-60-4759-1.
[75] J. Saramäki, M. Kivelä, J.-P. Onnela, K. Kaski, J. Kertesz, Generalizations of the clustering coefficient to weighted complex networks, Phys. Rev. E. 75 (2007) 27105.
[76] M. Oliveira, J. Gama, An overview of social network analysis, Wiley Interdiscip. Rev. Data Min. Knowl. Discov. 2 (2012) 99–115.
[77] T. Opsahl, F. Agneessens, J. Skvoretz, Node centrality in weighted networks: Generalizing degree and shortest paths, Soc. Netw. 32 (2010) 245–251.
[78] L.C. Freeman, Centrality in social networks conceptual clarification, Soc. Netw. 1 (1978) 215–239.
[79] T. Opsahl, V. Colizza, P. Panzarasa, J.J. Ramasco, Prominence and control: the weighted rich-club effect, Phys. Rev. Lett. 101 (2008) 168702.
[80] M. Bolanos, E.M. Bernat, B. He, S. Aviyente, A weighted small world network measure for assessing functional connectivity, J. Neurosci. Methods 212 (2013) 133–142, https://doi.org/10.1016/j.jneumeth.2012.10.004.
[81] M. Barthélemy, A. Barrat, R. Pastor-Satorras, A. Vespignani, Characterization and modeling of weighted networks, Phys. a Stat. Mech. Appl. 346 (2005) 34–43.
[82] M.E.J. Newman, The structure and function of complex networks, SIAM Rev. 45 (2003) 167–256.
[83] A.-L. Barabâsi, H. Jeong, Z. Néda, E. Ravasz, A. Schubert, T. Vicsek, Evolution of the social network of scientific collaborations, Phys. A Stat. Mech. Appl. 311 (2002) 590–614.
[84] G.F. Davis, M. Yoo, W.E. Baker, The small world of the American corporate elite 1982-2001, Strateg. Organ. 1 (2003) 301–326, https://doi.org/10.1177/14761270030013002.
[85] H. Ebel, L.-I. Mielsch, S. Bornholdt, Scale-free topology of e-mail networks, Phys. Rev. E. 66 (2002) 35103.
[86] P. Holme, C.R. Edling, F. Liljeros, Structure and time evolution of an internet dating community, Soc. Netw. 26 (2004) 155–174.
[87] P. Ingram, P.W. Roberts, Friendships among competitors in the sydney hotel industry1, Am. J. Sociol. 106 (2000) 387–423.
[88] M.E.J. Newman, Scientific collaboration networks II shortest paths, weighted networks, and centrality, Phys. Rev. E. 64 (2001) 16132.
[89] B. Uzzi, J. Spiro, Collaboration and creativity: The small world problem1, Am. J. Sociol. 111 (2005) 447–504.
[90] D.J. Watts, S.H. Strogatz, Collective dynamics of 'small-world'networks, Nature 393 (1998) 440–442.
[91] P.W. Holland, S. Leinhardt, Transitivity in structural models of small groups, Comp. Gr. Stud. (1971).
[92] T.A.B. Snijders, The statistical evaluation of social network dynamics, Sociol. Methodol. 31 (2001) 361–395.
[93] T.A.B. Snijders, P.E. Pattison, G.L. Robins, M.S. Handcock, New specifications for exponential random graph models, Sociol. Methodol. 36 (2006) 99–153.
[94] S.L. Feld, The focused organization of social ties, Am. J. Sociol. (1981) 1015–1035.
[95] M. Karlberg, Testing transitivity in graphs, Soc. Netw. 19 (1997) 325–343.
[96] M. Karlberg, Testing transitivity in digraphs, Sociol. Methodol. 29 (1999) 225–251.
[97] H. Louch, Personal network integration: transitivity and homophily in strong-tie relations, Soc. Netw. 22 (2000) 45–64.
[98] E. Ravasz, A.-L. Barabási, Hierarchical organization in complex networks, Phys. Rev. E. 67 (2003) 26112.
[99] E. Ravasz, A.L. Somera, D.A. Mongru, Z.N. Oltvai, A.-L. Barabási, Hierarchical organization of modularity in metabolic networks, Science (80-.) 297 (2002) 1551–1555.
[100] L. Lopez-Fernandez, G. Robles, J.M. Gonzalez-Barahona, Applying social network analysis to the information in CVS repositories, in: Int. Work. Min. Softw. Repos. 2004, pp. 101–105.
[101] G. Fagiolo, J. Reyes, S. Schiavo, On the topological properties of the world trade web: A weighted network analysis, Phys. A Stat. Mech. Appl. 387 (2008) 3868–3873.
[102] G. Kalna, D.J. Higham, A clustering coefficient for weighted networks, with application to gene expression data, Ai Commun. 20 (2007) 263–271.
[103] J.C. Reijneveld, S.C. Ponten, H.W. Berendse, C.J. Stam, The application of graph theoretical analysis to complex networks in the brain, Clin. Neurophysiol. 118 (2007) 2317–2331.
[104] M. Li, Y. Fan, J. Chen, L. Gao, Z. Di, J. Wu, Weighted networks of scientific communication: the measurement and topological role of weight, Phys. A Stat. Mech. Appl. 350 (2005) 643–656.
[105] P. Grindrod, Range-dependent random graphs and their application to modeling large small-world proteome datasets, Phys. Rev. E. 66 (2002) 66702.
[106] T. Opsahl, P. Panzarasa, Clustering in weighted networks, Soc. Netw. 31 (2009) 155–163.
[107] S. Wasserman, K. Faust, Social Network Analysis: Methods and Applications, Cambridge university press, 1994.
[108] S. Milgram, The small world problem, Psychol. Today. 2 (1967) 60–67.
[109] L. Katz, A new status index derived from sociometric analysis, Psychometrika. 18 (1953) 39–43.
[110] E.R. Peay, Connectedness in a general model for valued networks, Soc. Netw. 2 (1980) 385–410.
[111] S. Yang, D. Knoke, Optimal connections: strength and distance in valued graphs, Soc. Netw. 23 (2001) 285–295.
[112] E.W. Dijkstra, A note on two problems in connexion with graphs, Numer. Math. 1 (1959) 269–271.
[113] H.P. Thadakamalla, R. Albert, S.R.T. Kumara, Search in weighted complex networks, Phys. Rev. E. 72 (2005) 66128.
[114] X.-J. Xu, Z.-X. Wu, Y.-H. Wang, Properties of weighted complex networks, Int. J. Mod. Phys. C. 17 (2006) 521–529.
[115] U. Brandes, A faster algorithm for betweenness centrality, J. Math. Sociol. 25 (2001) 163–177.
[116] V. Latora, M. Marchiori, Efficient behavior of small-world networks, Phys. Rev. Lett. 87 (2001) 198701.
[117] V. Latora, M. Marchiori, Economic small-world behavior in weighted networks, Eur. Phys. J. B-Condens Matter Complex Syst. 32 (2003) 249–263.
[118] C.J. Stam, W. De Haan, A. Daffertshofer, B.F. Jones, I. Manshanden, A.M.V.C. Van Walsum, T. Montez, J.P.A. Verbunt, J.C. De Munck, B.W. Van Dijk, Graph theoretical analysis of magnetoencephalographic functional connectivity in Alzheimer's disease, Brain 132 (2009) 213–224.
[119] M.P. Stumpf, C. Wiuf, R.M. May, Subnets of scale-free networks are not scale-free: sampling properties of networks, Proc. Natl. Acad. Sci. USA (2005) 4221–4224.
[120] S.H. Lee, P.-J. Kim, H. Jeong, Statistical properties of sampled networks, Phys. Rev. E. 73 (2006) 016102, https://doi.org/10.1103/PhysRevE.73.016102.
[121] V. Krishnamurthy, M. Faloutsos, M. Chrobak, J.-H. Cui, L. Lao, A.G. Percus, Sampling large internet topologies for simulation purposes, Comput. Netw. 51 (2007) 4284–4302, https://doi.org/10.1016/j.comnet.2007.06.004.
[122] N.K. Ahmed, F. Berchmans, J. Neville, R. Kompella, Time-based sampling of social network activity graphs, in: Proc. Eighth Work. Min. Learn. with Graphs - MLG '10, ACM Press, New York, New York, USA, 2010, pp. 1–9, https://doi.org/10.1145/1830252.1830253.
[123] Y.-Y. Ahn, S. Han, H. Kwak, S. Moon, H. Jeong, Analysis of topological characteristics of huge online social networking services, in: Proc. 16th Int. Conf. World Wide Web, ACM, 2007, pp. 835–844.
[124] A. Mislove, M. Marcon, K.P. Gummadi, P. Druschel, B. Bhattacharjee, Measurement and analysis of online social networks, in: Proc. 7th ACM SIGCOMM Conf. Internet Meas, ACM, 2007, pp. 29–42.
[125] C. Wilson, B. Boe, A. Sala, K.P.N. Puttaswamy, B.Y. Zhao, User interactions in social networks and their implications, in: Proc. 4th ACM Eur. Conf. Comput. Syst, Acm, 2009, pp. 205–218.
[126] L.A. Goodman, Snowball sampling, Ann. Math. Stat. 32 (1961) 148–170, http://www.jstor.org/stable/2237615.
[127] J. Leskovec, J. Kleinberg, C. Faloutsos, Graphs over time, in: Proceeding Elev. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. - KDD '05, ACM Press, New York, New York, USA, 2005, p. 177, https://doi.org/10.1145/1081870.1081893.
[128] C.C. Aggarwal, On biased reservoir sampling in the presence of stream evolution, in: Proc. 32nd Int. Conf. Very Large Data Bases, 2006, pp. 607–618.
[129] S. Yoon, S. Lee, S.-H. Yook, Y. Kim, Statistical properties of sampled networks by random walks, Phys. Rev. E. 75 (2007) 046114, https://doi.org/10.1103/PhysRevE.75.046114.
[130] M. Gjoka, M. Kurant, C.T. Butts, A. Markopoulou, Walking in facebook: A case study of unbiased sampling of OSNs, in: Infocom, 2010 Proc. IEEE, IEEE, 2010, pp. 1–9.
[131] D.D. Heckathorn, Respondent-driven sampling: a new approach to the study of hidden populations, Soc. Probl. 44 (1997) 174–199.
[132] S. Goel, M.J. Salganik, Assessing respondent-driven sampling, Proc. Natl. Acad. Sci. 107 (2010) 6743–6747.
[133] A.S. Maiya, T.Y. Berger-Wolf, Sampling community structure, in: Proc. 19th Int. Conf. World Wide Web, ACM, 2010, pp. 701–710.
[134] M. Kurant, M. Gjoka, C.T. Butts, A. Markopoulou, Walking on a graph with a magnifying glass, in: Proc. ACM SIGMETRICS Jt. Int. Conf. Meas. Model. Comput. Syst. - SIGMETRICS '11, ACM Press, New York, New York, USA, 2011, p. 281, https://doi.org/10.1145/1993744.1993773.

[135] A.H. Rasti, M. Torkjazi, R. Rejaie, N. Duffield, W. Willinger, D. Stutzbach, Respondent-driven sampling for characterizing unstructured overlays, in: INFOCOM 2009, IEEE, IEEE, 2009, pp. 2701–2705.
[136] D. Stutzbach, R. Rejaie, N. Duffield, S. Sen, W. Willinger, On unbiased sampling for unstructured peer-to-peer networks, IEEE/ACM Trans. Netw. 17 (2009) 377–390.
[137] K. Avrachenkov, B. Ribeiro, D. Towsley, Improving random walk estimation accuracy with uniform restarts, in: Int. Work. Algorithms Model. Web-Graph, Springer, 2010, pp. 98–109.
[138] J. Lu, D. Li, Sampling online social networks by random walk, in: Proc. First ACM Int. Work. Hot Top. Interdiscip. Soc. Networks Res. - HotSocial '12, ACM Press, New York, New York, USA, 2012, pp. 33–40, https://doi.org/10.1145/2392622.2392628.
[139] M. Kurant, A. Markopoulou, P. Thiran, Towards unbiased BFS sampling, IEEE J. Sel. Areas Commun. 29 (2011) 1799–1809, https://doi.org/10.1109/JSAC.2011.111005.
[140] A.-L. Barabási, R. Albert, Emergence of scaling in random networks, Science (80-.) 286 (1999) 509–512.
[141] R. Albert, H. Jeong, A.-L. Barabási, Error and attack tolerance of complex networks, Nature 406 (2000) 378–382.
[142] K.S. Narendra, M.A.L. Thathachar, Learning Automata: An Introduction, Courier Corporation, 2012, https://doi.org/10.1109/TSMCB.2002.1049606.
[143] M.A.L. Thathachar, P.S. Sastry, Networks of Learning Automata: Techniques for Online Stochastic Optimization, Springer Science & Business Media, 2011.
[144] Alireza Rezvanian, et al., Social Network Sampling, Learning Automata Approach for Social Networks, 820, Springer, Cham, 2019, http://dx.doi.org/10.1007/978-3-030-10767-3_4.
[145] M. Ghavipour, M.R. Meybodi, An adaptive fuzzy recommender system based on learning automata, Electron. Commer. Res. Appl. 20 (2016) 105–115, https://doi.org/10.1016/j.elerap.2016.10.002.
[146] Alireza Rezvanian, et al., Social Recommender Systems, Learning Automata Approach for Social Networks, 820, Springer, Cham, 2019, http://dx.doi.org/10.1007/978-3-030-10767-3_8.
[147] Alireza Rezvanian, et al., Social Trust Management, Learning Automata Approach for Social Networks, 820, Springer, Cham, 2019, http://dx.doi.org/10.1007/978-3-030-10767-3_7.
[148] Alireza Rezvanian, et al., Social Link Prediction, Learning Automata Approach for Social Networks, 820, Springer, Cham, 2019, http://dx.doi.org/10.1007/978-3-030-10767-3_6.
[149] Alireza Rezvanian, et al., Social Community Detection, Learning Automata Approach for Social Networks, 820, Springer, Cham, 2019, http://dx.doi.org/10.1007/978-3-030-10767-3_5.
[150] Alireza Rezvanian, et al., Social Influence Maximization, Learning Automata Approach for Social Networks, 820, Springer, Cham, 2019, http://dx.doi.org/10.1007/978-3-030-10767-3_9.
[151] M.A.L. Thathachar, K.M. Ramachandran, Asymptotic behaviour of a learning algorithm, Int. J. Control. 39 (1984) 827–838.
[152] F. Harary, G. Gupta, Dynamic graph models, Math. Comput. Model. 25 (1997) 79–87.
[153] J.A. Torkestani, M.R. Meybodi, Finding minimum weight connected dominating set in stochastic graph based on learning automata, Inf. Sci. (Ny). 200 (2012) 57–77.
[154] S. Misra, B.J. Oommen, Dynamic algorithms for the shortest path routing problem: learning automata-based solutions, IEEE Trans. Syst. Man, Cybern. Part B. 35 (2005) 1179–1192.
[155] H. Beigy, M.R. Meybodi, Utilizing distributed learning automata to solve stochastic shortest path problems, Int. J. Uncertain. Fuzziness Knowl.-Based Syst. 14 (2006) 591–615.
[156] M.S. Granovetter, The strength of weak ties, Am. J. Sociol. (1973) 1360–1380.
[157] A. Jøsang, E. Gray, M. Kinateder, Simplification and analysis of transitive trust networks, Web Intell. Agent Syst. An Int. J. 4 (2006) 139–161.
[158] M. Richardson, R. Agrawal, P. Domingos, Trust management for the semantic web, in: Int. Semant. Web Conf, Springer, 2003, pp. 351–368.
[159] W. Jiang, J. Wu, G. Wang, On selecting recommenders for trust evaluation in online social networks, ACM Trans. Internet Technol. 15 (2015) 14, https://doi.org/10.1145/2807697.
[160] G. Cimini, T. Squartini, F. Saracco, D. Garlaschelli, A. Gabrielli, G. Caldarelli, The statistical physics of real-world networks, Nat. Rev. Phys. 1 (2019) 58–71, https://doi.org/10.1038/s42254-018-0002-6.