Information Diffusion on Temporal Networks

Zhan, X.

DOI
10.4233/uuid:25b01015-559c-418b-b52e-0c92a6b84531

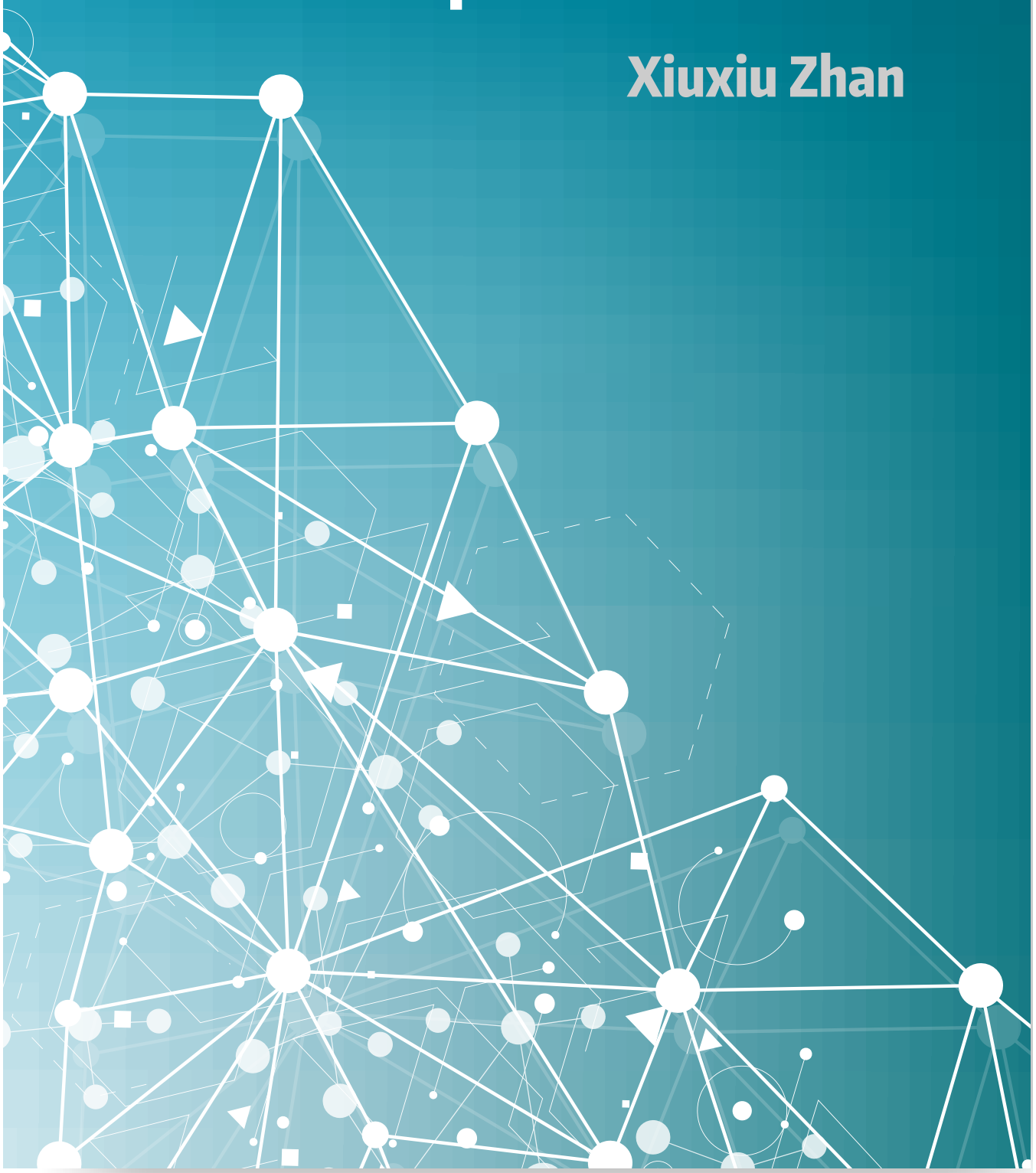
Publication date
2020

Document Version
Final published version

Citation (APA)
Zhan, X. (2020). *Information Diffusion on Temporal Networks*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:25b01015-559c-418b-b52e-0c92a6b84531

Information Diffusion on Temporal Networks

Xiuxiu Zhan



INFORMATION DIFFUSION ON TEMPORAL NETWORKS

INFORMATION DIFFUSION ON TEMPORAL NETWORKS

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. dr. ir. T.H.J.J. van der Hagen,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op
woensdag 7 oktober 2020 om 12:30 uur

door

Xiuxiu ZHAN

Master of Science in Mathematics,
North University of China, Taiyuan, China,
geboren te Anhui, China.

Dit proefschrift is goedgekeurd door de

promotor: Prof. dr. A. Hanjalic
copromotor: Dr. ir. H. Wang

Samenstelling promotiecommissie:

Rector Magnificus	voorzitter
Prof. dr. A. Hanjalic	Technische Universiteit Delft
Dr. ir. H. Wang	Technische Universiteit Delft

Onafhankelijke leden:

Prof. N. Litvak	Universiteit Twente
Prof. P. Holme	Tokyo Institute of Technology, Japan
Prof. dr. Y. Moreno	University of Zaragoza, Spain
Prof. dr. ir. A. Bozzon	Technische Universiteit Delft
Prof. dr. ir. R.E. Kooij	Technische Universiteit Delft

Keywords: Temporal Networks, Information Diffusion, Network Representation Learning, Link Prediction, Node Classification
Printed by: Ridderprint BV

ISBN 978-94-6416-144-1

An electronic version of this dissertation is available at
http://repository.tudelft.nl/.

To my family

CONTENTS

SUMMARY

As an important carrier of information diffusion, social media has experienced a huge increase in the number of users and also has a big effect on the way of how information diffuses. For example, Facebook and Youtube have attracted more than 1.6 and 1.3 billion users until 2020, respectively. The use of internet and online social network have largely reduced the cost of information propagation and sharing. Besides users and content-based features, social network properties are critical factors that may affect information diffusion. In this thesis, we focus on the influence of temporal network properties on information spreading. As researchers have proved that similar users tend to spread similar content of information, we further propose how to design network representation learning algorithms to better capture node similarity in a network.

The first part of the thesis is mainly about how the local properties of nodes and links would affect information spreading on temporal networks. Chapter 2 studies which links are likely to appear in an information diffusion trajectory. We simulate the information diffusion process by a susceptible-infected (SI) model on various empirical temporal networks. An information diffusion backbone is proposed to characterize the probability of a link to appear in the diffusion trajectory. Due to the high complexity of constructing diffusion backbone, we further propose time-scaled weight to identify which links would appear in the diffusion backbone. Compared to the centrality metrics derived from static networks, time-scaled weight shows better identification performance. The conclusions in this chapter may inspire how to maximize information diffusion on temporal networks by deliberately choosing links to transmit information. Chapter 3 investigates which links should be temporally blocked in order to suppress information diffusion on temporal networks. We rank the links by different blocking strategies based on the link properties on static and temporal networks, including the ones derived from information diffusion backbone. We remove the links with high ranking values based on blocking strategies for a given time period. We show that four link blocking strategies outperform the others in suppressing information diffusion. The results show that the effectiveness of the metrics on suppressing information diffusion largely depends on the network properties. In chapter 4, we study how to identify influential nodes, i.e., nodes serving as the seed can spread information widely, on temporal networks. The information diffusion process is simulated by susceptible-infected-recovered (SIR) model on various empirical temporal networks. We propose a temporal information gathering process (*Tig*-process), which can iteratively gather neighboring information though temporal path, to identify influential nodes. Compared to the benchmark metrics, *Tig*-process can better identify influential nodes across different temporal networks with a small cost. The experimental designs and results in these three chapters further inspire us to study the local surrounding properties of nodes and links for other spreading processes as well as other types of networks.

In the second part of the thesis, we work on designing network embedding algorithms to embed nodes to a low-dimensional space, which can make similar nodes be close in the embedding space. Chapter 5 designs a degree-biased random walk, i.e., *DiaRW*, to sample walks from a static network. If the source node of a random walk has higher degree, the walk length tends to be longer. Also, if a random walker walks to a low-degree node, the

probability of backtracking the former high-degree node is higher. The node pairs generated from walks are further used as input for a learning model, i.e., Skip-Gram model. We unveil that *DiaRW* shows better performance compared to baseline embedding algorithms on tasks, e.g., link prediction and node classification. Chapter 6 proposes SI-spreading-based network embedding algorithms. We apply SI model on static and temporal networks to sample trajectories. The node pairs generated from trajectories are also used as input for Skip-Gram model. We show SI-spreading-based network embedding algorithms perform better than random-walk-based network embedding algorithms on missing link prediction task. Both of the two chapters consider node heterogeneity in designing embedding algorithms.

The last chapter proposes insight of the thesis based on the research questions and provides the possible future directions that is related to our research.

1

INTRODUCTION

We should be taught not to
wait for inspiration to start a thing.
Action always generates inspiration.
Inspiration seldom generates action.

Frank Tibolt

Any intelligent fool can make things
bigger, more complex, and more violent.
It takes a touch of genius –
and a lot of courage –
to move in the opposite direction.

EF Schumacher

1

1.1. BACKGROUND

The development of information society and Internet technology has largely increased the use of online social platforms, such as Facebook, Twitter and WeChat, in the population all over the world. In the past, one could say that two individuals are separated by only six other people on average, which is known as *the six degrees of separation* [1]. However, the introduction of social media has significantly reduced the distance between people. Edunov et al. [2] show that the average distance between two users on Facebook is 3.57. This enables easy and quick communication and enables information spreading at the pace much higher than in the past. Additionally, we increasingly rely on online information because of the quick and easy access to information from social media, which is available in the form of text, images, video and websites of various types, such as product recommendation, news and political opinions. Low threshold for posting and spreading information via social media not only contributes to making us better informed, but unfortunately also gets misused for spreading fake or misleading information. The diffusion of real information should be encouraged as it is a way to share knowledge and contribute to the advancement of the society. The spread of misinformation may, however, introduce confusion, complicate public's understanding of situations, events and public policies, and contribute to a wrong bias in forming beliefs and for attitude formation. This could lead to undesired effects, such as societal polarization and segregation. It is for this reason that the World Economic Forum (WEF) has listed massive digital misinformation as one of the main threats to human society [3, 4]. It is thus crucial to find effective ways to suppress the spreading of misinformation, while accelerating the diffusion of the real (useful) information. In order to achieve this goal, we first need to understand well the mechanisms underlying information spreading (diffusion) via online social platforms. Such understanding will not only help us rethink the information spreading, but also equip us with powerful means to control other spreading processes, like epidemic spreading or spreading of computer viruses.

Information spreads through interactions between individuals. Such interactions can be represented as a *network*, where *nodes* represent individuals and *links* represent their interactions. In other words, a link exists if there is an interaction between two nodes. Since interactions are typically time-dependent, so are the links between the nodes connected to (and thus valid at) the time stamps at which they occur. In this case, we speak about a *temporal network*. It is different from a *static network*, which captures only which node pairs have (had) interaction(s) in the past and/or how many interactions they have (had). As long as two nodes have had at least one interaction, a link is formed in the corresponding static network. This opens the way for aggregating over time the link occurrence and disappearance processes in a temporal network into an 'integral' representation using a static network. The time ordering of the contacts and the correlations between contacts, however, can only be captured and analyzed by using a temporal network representation, making this representation most suitable for analyzing information spreading processes.

To understand how information spreads on a temporal network, we need to consider the following two aspects, i.e., how to model the information diffusion process and how to study the effect of network *topology*, a constellation of nodes and links, on that process. Stochastic models, especially data-driven models, have emerged to describe how information spreads on online social platforms. In such models, one assumes that nodes have different states. For example, an individual either knows (thus possesses) the information or does not know the information. This state of a node may change over time through interactions with other nodes in a network, for example, by forwarding a message. To characterize these states, it is

common to rely on the terminology used by classical epidemic models, such as *susceptible-infected* (SI), *susceptible-infected-susceptible* (SIS) and *susceptible-infected-recovered* (SIR) models, *independent cascade model*, *threshold model* and *opinion model* [5, 6]. We take SI and SIR models as examples to illustrate how information diffusion models work. In the SI model, each node is in one of the two states, i.e., susceptible or infected, at any time. A node is in the susceptible state if the corresponding individual does not know the information, but is open to receive it (thus 'susceptible') through an information sharing interaction. A node is in the infected state if the corresponding individual is 'infected' (thus possesses) by the information. An infected node infects a susceptible neighbor with a given infection (information transmission) probability when the two nodes have an interaction. In the SIR model, the extra 'recovered' state is added. This is the state of the node after 'forgetting' the information it was 'infected' with before. Being in this state, a node can neither be infected nor infect any other node anymore.

In a temporal network, the network properties, like for example, the time ordering of contacts between nodes, would affect the information diffusion process. The daily and circadian rhythms of human interactions influencing this ordering may therefore directly impact the information transmission paths. Furthermore, temporal networks have been shown to manifest seemingly universal properties, such as burstiness[1]. One of the key questions in analyzing information spreading mechanisms is how the properties of the underlying temporal network affect the *information prevalence* (diffusion size), i.e., the number of nodes that have ever possessed the information. For example, Karsai et al. [7] have shown that the burstiness property of temporal networks can slow down the SI spreading process. In this thesis, we broaden the analysis of the influence of a temporal network on information spreading by focusing on more network properties.

1.2. THESIS SCOPE AND CONTRIBUTION

When information spreads on a temporal network, the nodes and links may have different roles in a spreading process. For instance, not all links would appear in a diffusion trajectory thus actually spread the information from one node to another. Even the links or nodes that appear in an information diffusion process may show different importance for that process. As a consequence, information originating from different nodes may result in different final diffusion sizes. This is because nodes and links are heterogeneous in topological and temporal properties, such as node *degree*, the number of contacts that a node has, and the waiting time between two contacts [8–11]. The number of contacts that a node has in a temporal network has been shown to follow a power-law distribution, with most nodes having a few contacts and a few nodes having a large number of contacts [12]. Nodes that have more frequent contacts with other nodes tend to have higher probability to transmit information to other nodes. If a node seldom has contacts with others, it is difficult for information to spread out from it. Previous work has studied how the statistical properties, such as the distribution of the number of contacts, influence a spreading process on a temporal network [7, 13]. The fundamental question that has not been studied before and will be investigated in this thesis is, *how local properties of nodes and links are associated with their roles in facilitating or suppressing information diffusion?*

[1] The tendency that contacts of a node occur in bursts within a short time duration and such intermittent active periods are separated by long inactive ones. A power-law distribution of the waiting time between consecutive contacts has been widely observed [7].

We study this problem on a temporal network in the following three steps. Firstly, we investigate which links are more likely to contribute to the actual information diffusion process, i.e., appear in a diffusion trajectory. Secondly, we study which links, in view of their properties, should be removed to suppress information diffusion. Thirdly, we explore which node, again in view of its properties, to choose as a seed node to start an information diffusion process to make the information spread as widely as possible. These three steps serve as fundamental investigation of information diffusion and can be applied to maximization and control of information diffusion.

From another perspective, based on their investigation of information diffusion on Facebook, Vicario et al. [4] found that nodes with similar properties tend to spread similar sort of information. In view of this finding, evaluating the similarity of network nodes may help identify the nodes that spread misinformation and block them to suppress the spreading. To assess node similarity, traditionally, the network adjacency matrix was used. The elements of the matrix indicate whether pairs of nodes are adjacent or not in the network. This simple representation or embedding displays, however, only first-order relationships between nodes, but not high-order structures. Recently, *network representation learning* (NRL), which can capture high-order relationships between nodes, has been proposed to learn low-dimensional embedding vectors for nodes, while preserving network topology structure, node content and other side information. The goal is to bring the nodes with similar properties at a close proximity to each other in the learned representation space. Inspired by this possibility, the second problem addressed in this thesis is *how to design NRL algorithms that better capture the similarity between nodes in a network.*

Among the representation learning algorithms proposed recently, random-walk-based network embedding algorithms have shown good performance to embed a network [14, 15]. Random-walk-based network embedding algorithms utilize random walk to sample the network structure. The node pairs generated from the random walk trajectory paths are further used as the input for a Skip-Gram model, a representative language model that embeds nodes into vectors. Previous random-walk-based network embedding algorithms, such as *DeepWalk* and *Node2Vec* [14, 15], sample equal number of trajectory paths and equal length of the trajectory paths for every node in the network. They have not considered the node heterogeneity when proposing the sampling strategy, resulting in a lot of repeated node pairs (i.e., redundant information) as the input for Skip-Gram model. Also, these algorithms are difficult to be applied to large networks with millions of nodes. To propose scalable algorithms, we start from investigating how to consider node properties in designing random-walk-based network embedding algorithms. In addition, we explore how to utilize information diffusion process to replace random walk process to sample the trajectory paths for network embedding. The embedding vectors learned from the NRL algorithms can be further applied to network analytic tasks, such as node classification, community detection and link prediction, possibly in combination with conventional vector-based machine learning algorithms.

1.3. THESIS OBJECTIVES AND OUTLINE

Regarding the two general problems we proposed in the previous section, we now map them onto a number of research questions and explain how they are addressed in different chapters of the thesis. This mapping is illustrated in Figure 1.1.

Chapter 2, 3 and **4** concentrate on the first problem of the thesis, namely the investigation of the influence of the properties of the nodes and links on information diffusion processes on

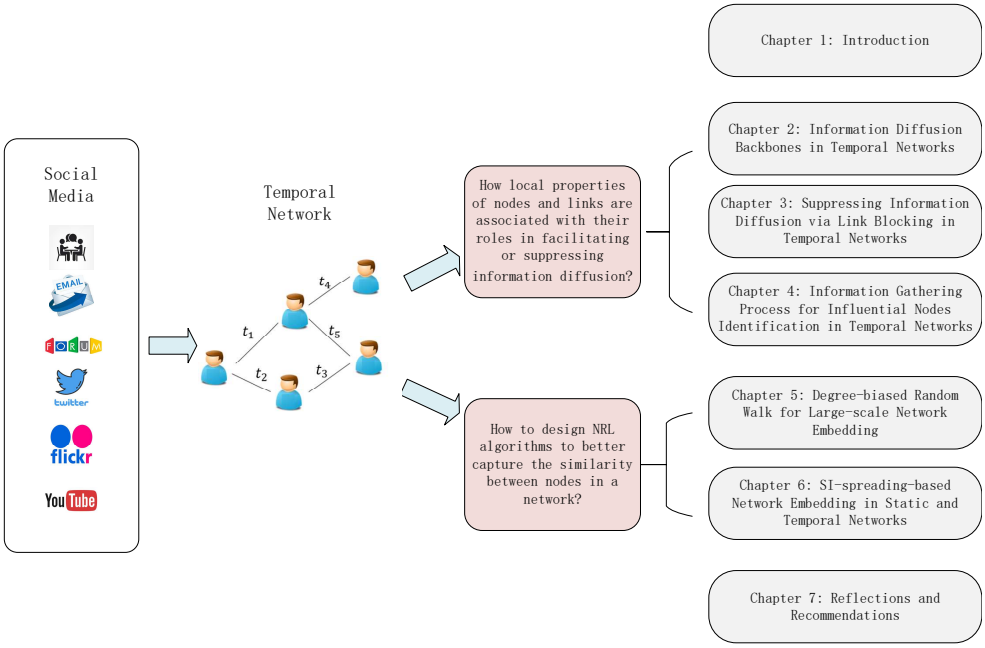


Figure 1.1: The structure of the thesis.

temporal networks. Specifically, **Chapter 2** studies the influence of properties surrounding a link on information diffusion through that link. The susceptible-infected (SI) spreading process on empirical temporal networks is considered with the aim to answer the following research questions:

- If a piece of information diffuses on a temporal network, which links are likely to appear in the diffusion trajectory?
- How can we identify links that frequently appear in a diffusion trajectory?

The study of which links are likely to contribute to the actual diffusion of information may contribute to the prediction of information trajectory if a new piece of information spreads on a temporal network. In this chapter, we propose an information diffusion backbone to characterize the likelihood of a link to appear in an SI spreading process.

The suppression of misinformation spreading is crucial. This can be realized by removing or blocking temporal links from a network. A key research question here is

- Which temporal links should be removed to suppress information spreading?

Chapter 3 proposes strategies to find crucial links to remove for this purpose. We focus on how different local properties would help to determine a link's role in suppressing SI diffusion process on a temporal network, including information diffusion backbone proposed in **Chapter 2**.

An influential node for information diffusion is defined as a node that, if serving as the seed node, could spread the information widely. Influential node identification has attracted

increasing attention lately, as it helps to control the spread of misinformation or epidemic, to promote the diffusion of real information or news, and to conduct successful advertisements for viral marketing as well as to stop catastrophic outages in power grids and the Internet [16–18]. **Chapter 4** studies the problem of how to identify or rank influential nodes in a temporal network via temporal and topological properties of a node. The information diffusion process is modeled by the SIR model in this chapter. We aim at answering the following research questions:

- If we aim to maximize the diffusion of a piece of information, which node should we choose as the seed node?
- Which topological and temporal properties of nodes can be used to identify such influential seed nodes?
- How to evaluate the effectiveness of the influential node identification methods based on diverse topological and temporal nodal properties?

Chapters 5 and **6** address the second problem of the thesis and focus on network embedding algorithms by considering node heterogeneity, while designing sampling strategies, e.g., random-walk-based and SI-spreading-based sampling strategies. As the first step, we discuss the limitations of previous proposed random-walk-based network embedding algorithms and propose an efficient random-walk-based network embedding algorithm. When we design random walk to sample trajectory paths for each node, we assume that nodes with different importance have different number of trajectory paths as well as different lengths of the paths. In **Chapter 5**, we aim to answer the following research questions:

- Which node centrality metric should we use in designing random-walk-based network embedding algorithm?
- Does the embedding algorithm we propose perform better than the state-of-the-art algorithms?
- How is the performance of the algorithm dependent on the properties of the network, i.e., in what kind of network does the embedding algorithm we propose perform better?

Since the previous network embedding algorithms are random-walk-based, we are curious about using other spreading processes, such as SI model, to sample the network structure. In **Chapter 6**, we investigate how to design network embedding algorithms based on SI spreading processes in both static and temporal networks. The node embedding vectors are further used for link prediction. **Chapter 6** answers the following research questions:

- How to utilize an SI spreading process to embed networks? How is the performance of SI-spreading-based network embedding in comparison with random-walk-based ones?
- If SI-spreading-based network embedding algorithms outperform the state-of-the-art, where does this improvement come from?
- Which of our algorithms performs better: static network embedding or temporal network embedding? Can temporal information help to improve the performance?

Chapter 7 highlights the contributions of this thesis and points out possible future directions.

1.4. PUBLICATION RELATED TO THIS THESIS

The following papers are completed by the author of this thesis while pursuing the Ph.D. degree at Delft University of Technology.

1. **X.-X. Zhan**, A. Hanjalic and H. Wang, *Information Diffusion Backbones in Temporal Networks*, Scientific Reports **9(1)**, 6798 (2019). **[Chapter 2]**

2. **X.-X. Zhan**, A. Hanjalic and H. Wang, *Suppressing Information Diffusion via Link Blocking in Temporal Networks*, In International Conference on Complex Networks and Their Applications, Springer, Cham. 448-458 (2019). **[Chapter 3]**

3. C. Qu, **X.-X. Zhan***, G. Wang, J. Wu and Z.-K. Zhang, *Temporal Information Gathering Process for Node Ranking in Time-varying Networks*, Chaos: An Interdisciplinary Journal of Nonlinear Science, **29(3)**, 033116 (2019). **[Chapter 4]**

4. Y. Zhang, Z Shi, D Feng and **X.-X. Zhan***, *Degree-biased Random Walk for Large-scale Network Embedding*, Future Generation Computer Systems, **100**, 198-209 (2019). **[Chapter 5]**

5. **X.-X. Zhan**, Z. Li, N. Masuda, P. Holme and H. Wang, *SI-spreading-based Network Embedding in Static and Temporal Networks*, Submitted to EPJ Data Science. **[Chapter 6]**

We note that **Chapter 4** and **5** are based on the publications where the PhD candidate is not the first author. These publications are, however, included in the thesis because of the significant contribution of the candidate regarding the main idea, experimental design and the writing of the paper. As a further note, for **Chapter 4**, we use a part of the underlying publication and revise the content to make it fit the thesis.

1.5. HOW TO READ THIS THESIS

Chapter 2, 3, 4 and 6 in this thesis adopt original publications. We give the references of the corresponding publications in the footnote at the beginning of each chapter. Each chapter serves as an independent work and can be read without linking to the previous chapters. The notations and terminologies may differ in different chapters. Because we publish the papers in different scientific journals or conferences, the length and depth of chapters may vary in different chapters. Additionally, the chapters that address similar topics may share similar motivations, arguments and materials.

1

REFERENCES

[1] J. Guare, *Six degrees of separation: A play* (Vintage, 1990).

[2] S. Edunov, C. Diuk, I. O. Filiz, S. Bhagat, and M. Burke, *Three and a half degrees of separation,* Research at Facebook (2016).

[3] L. Howell *et al.*, *Digital wildfires in a hyperconnected world,* WEF report **3**, 15 (2013).

[4] M. Del Vicario, A. Bessi, F. Zollo, F. Petroni, A. Scala, G. Caldarelli, H. E. Stanley, and W. Quattrociocchi, *The spreading of misinformation online,* Proceedings of the National Academy of Sciences **113**, 554 (2016).

[5] A. Vespignani, *Modelling dynamical processes in complex socio-technical systems,* Nature physics **8**, 32 (2012).

[6] Z.-K. Zhang, C. Liu, X.-X. Zhan, X. Lu, C.-X. Zhang, and Y.-C. Zhang, *Dynamics of information diffusion and its applications on complex networks,* Physics reports **651**, 1 (2016).

[7] M. Karsai, M. Kivelä, R. K. Pan, K. Kaski, J. Kertész, A.-L. Barabási, and J. Saramäki, *Small but slow world: How network topology and burstiness slow down spreading,* Physical review E **83**, 025102 (2011).

[8] E. C. Costa, A. B. Vieira, K. Wehmuth, A. Ziviani, and A. P. C. Da Silva, *Time centrality in dynamic complex networks,* Advances in Complex Systems **18**, 1550023 (2015).

[9] H. Kim and R. Anderson, *Temporal node centrality in complex networks,* Physical Review E **85**, 026107 (2012).

[10] T. Takaguchi, N. Sato, K. Yano, and N. Masuda, *Importance of individual events in temporal networks,* New Journal of Physics **14**, 093003 (2012).

[11] D. Taylor, S. A. Myers, A. Clauset, M. A. Porter, and P. J. Mucha, *Eigenvector-based centrality measures for temporal networks,* Multiscale Modeling & Simulation **15**, 537 (2017).

[12] N. Masuda and P. Holme, *Temporal network epidemiology* (Springer, 2017).

[13] R. Lambiotte, L. Tabourier, and J.-C. Delvenne, *Burstiness and spreading on temporal networks,* The European Physical Journal B **86**, 320 (2013).

[14] B. Perozzi, R. Al-Rfou, and S. Skiena, *Deepwalk: Online learning of social representations,* in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD'14 (ACM, New York, NY, USA, 2014) pp. 701–710.

[15] A. Grover and J. Leskovec, *node2vec: Scalable feature learning for networks,* in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining* (ACM, 2016) pp. 855–864.

[16] P. R. Soares and R. B. Prudêncio, *Proximity measures for link prediction based on temporal events,* Expert Systems with Applications **40**, 6652 (2013).

[17] L. Tabourier, A.-S. Libert, and R. Lambiotte, *Predicting links in ego-networks using temporal information,* EPJ Data Science **5**, 1 (2016).

[18] L. Lü, D. Chen, X.-L. Ren, Q.-M. Zhang, Y.-C. Zhang, and T. Zhou, *Vital nodes identification in complex networks,* Physics Reports **650**, 1 (2016).

2

INFORMATION DIFFUSION BACKBONES IN TEMPORAL NETWORKS

This chapter have been published as: X.-X. Zhan, A. Hanjalic and H. Wang, Information Diffusion Backbones in Temporal Networks, Scientific Reports **9(1)**, 6798 (2019).

In this work, we explore: which node pairs are likely to contribute to the actual diffusion of information, i.e., appear in a diffusion trajectory? How is this likelihood related to the local temporal connection features of the node pair? Such deep understanding of the role of node pairs is crucial to tackle challenging optimization problems such as which kind of node pairs or temporal contacts should be stimulated in order to maximize the prevalence of an information spread. We start by using Susceptible-Infected (SI) model, in which an infected (information possessing) node could spread the information to a susceptible node with a given infection probability β whenever a contact happens between the two nodes, as the information diffusion process. We consider a large number of real-world temporal networks. First, we propose the construction of an information diffusion backbone *$G_B(\beta)$ for a SI spreading process with an infection probability β on a temporal network. The backbone is a weighted network where the weight of each node pair indicates how likely the node pair appear in a diffusion trajectory starting from an arbitrary node. Second, we investigate the relation between the backbones with different infection probabilities on a temporal network. We find that the backbone topology obtained for low and high infection probabilities approach the backbone $G_B(\beta \to 0)$ and $G_B(\beta = 1)$, respectively. The backbone $G_B(\beta \to 0)$ equals the integrated weighted network, where the weight of a node pair counts the total number of contacts in between. Finally, we explore node pairs with what local connection features tend to appear in $G_B(\beta = 1)$, thus actually contribute to the global information diffusion. We discover that a local connection feature among many other features we proposed, could well identify the (high-weight) links in $G_B(\beta = 1)$. This local feature encodes the time that each contact occurs, pointing out the importance of temporal features in determining the role of node pairs in a dynamic process.*

2.1. INTRODUCTION

Both online social networks like Facebook, Twitter and LinkedIn and physical contact networks facilitate the diffusion of information where a piece of information is transmitted from one individual to another through their online or physical contacts or interactions. Information diffusion processes have been modeled by, e.g., independent cascade models [1], threshold models [2] and epidemic spreading models [3–6]. Social networks have been first considered to be static where nodes represent the individuals and links indicate the relation between nodes such as whether they have ever contacted or not [7]. Information is assumed to propagate through the static links according to the aforementioned models. Recently, the temporal nature of contact networks has been taken into account in the spreading processes, i.e., the contacts between a node pair occur at specific time stamps (the link between nodes is time dependent) and information could possibly propagate only through contacts (or temporal links) [8–12]. Consider the SI (Susceptible-Infected) spreading process on a temporal network [3, 5]. Each individual can be in one of the two states: susceptible (S) or infected (I). A node in the infected (susceptible) state means that it has (does not have) the information. A susceptible node could get infected with an infection probability β via each contact with an infected node. An infected individual remains infected forever.

Progress has been made in the exploration of how temporal network features [13–17] and the choice of the source node [18, 19] influence a diffusion process especially its diffusion size, i.e., the number of nodes reached. However, we lack foundational understanding of which kind of node pairs are likely to contribute to an actual information diffusion process, i.e., appear in an information diffusion trajectory. Such understanding is essential to explain

and control the prevalence of information spread (e.g., which node pairs should be stimulated to contact at what time in order to maximize the prevalence?). The contact frequency between nodes, as typically used in static networks, is not the only factor that would affect the appearance of a node pairs in an information diffusion trajectory, as we need to consider the time stamps of the contacts as well [20–23]. For instance, the node pairs with a lot of contacts that only happened before the information starts to diffuse are of no importance for the diffusion process.

In this chapter, we address the question of which kind of node pairs are likely to contribute to the diffusion of information, considering the SI diffusion process as a start. Specifically, we explore how the probability that a node pair appears in a diffusion trajectory is related to local temporal connection features of the two nodes. First, we propose the construction of an *information diffusion backbone* $G_B(\beta)$ for a SI spreading process with an infection probability β on a given temporal network. The construction is based on a large number of information diffusion trajectories. The resultant backbone is a weighted network where the weight of each node pair indicates how likely the node pair contributes to a diffusion process that starts from an arbitrary node. We consider a large number of empirical temporal networks. For each network, we construct diffusion backbones for diverse infection probabilities and study the relationship between these backbones. We find that backbone topology varies from $G_B(\beta = 0) \triangleq G_B(\beta \to 0)$ (which equals the integrated weighted network) when the spreading probability β is small to $G_B(\beta = 1)$ when the infection probability is large. The difference between the two extreme backbones $G_B(\beta = 0)$ and $G_B(\beta = 1)$, suggests the extent to which the backbones with diverse infection rates may vary. Finally, we investigate further which local connection feature of a node pair may suggest its high weight in the backbone $G_B(\beta = 1)$. One of the features that we proposed incorporates only the time stamps when contacts occur between a node pair. It outperforms other classic features of a node pair including those derived from the integrated network, which points out the importance of temporal information in determining the role of a node pair in a diffusion process. The computational complexity of $G_B(\beta = 1)$ is high. Our finding of the relation between local temporal features of a node pair and its global contribution to an information diffusion allows the approximation of the information backbone by computing a local temporal feature that is of low computational complexity.

The chapter is organized as follows. In Section 2.2, we first introduce how to represent a temporal network and then explain the process of constructing the information diffusion backbone for a SI diffusion process on a temporal network. Finally, we illustrate a set of empirical temporal networks that will be used in the following experiments. In Section 2.3, we present our comparative analysis of the constructed backbones for different infection probabilities and for different networks. At the end of this section, we evaluate which local connection features of a node pair, including the measures we proposed, can identify whether the node pair will be connected in the backbone $G_B(\beta = 1)$ and with a high weight or not. A discussion concludes the chapter in Section 2.4.

2.2. MATERIALS AND METHODS

2.2.1. REPRESENTATION OF A TEMPORAL NETWORK

A temporal network can be measured by observing the contacts between each node pair at each time step within a given time window $[0, T]$ and represented as $\mathcal{G} = (\mathcal{N}, \mathcal{L})$. Here, $\mathcal{N}$ is the node set, with the size $N = |\mathcal{N}|$ representing the number of nodes in the network, and

$\mathscr{L} = \{l(j,k,t), t \in [0,T], j,k \in \mathscr{N}\}$ is the contact set, where the element $l(j,k,t)$ indicates that the nodes j and k have a contact at time step t. A temporal network can also be described by a three-dimensional binary adjacency matrix $\mathscr{A}_{N\times N\times T}$, where the elements $\mathscr{A}(j,k,t) = 1$ and $\mathscr{A}(j,k,t) = 0$ represent, respectively, that there is a contact or no contact between the nodes j and k at time step t.

An integrated weighted network $G_W = (\mathscr{N}, \mathscr{L}_W)$ can be derived from a temporal network $\mathscr{G}$ by aggregating the contacts between nodes over the entire observation time window T. In other words, two nodes are connected in G_W if there is at least one contact between them in $\mathscr{G}$. Each link $l(j,k)$ in $\mathscr{L}_W$ is associated with a weight w_{jk} counting the total number of contacts between node j and k in $\mathscr{G}$. The integrated weighted network G_W can therefore be described by a weighted adjacency matrix $A_{N\times N}$, with its element

$$A(j,k) = \sum_{t=1}^{T} \mathscr{A}(j,k,t) \tag{2.1}$$

counting the number of contacts between a node pair. An example of a temporal network $\mathscr{G}$ and its integrated weighted network G_W are given in Figure 2.1(a) and (b), respectively.

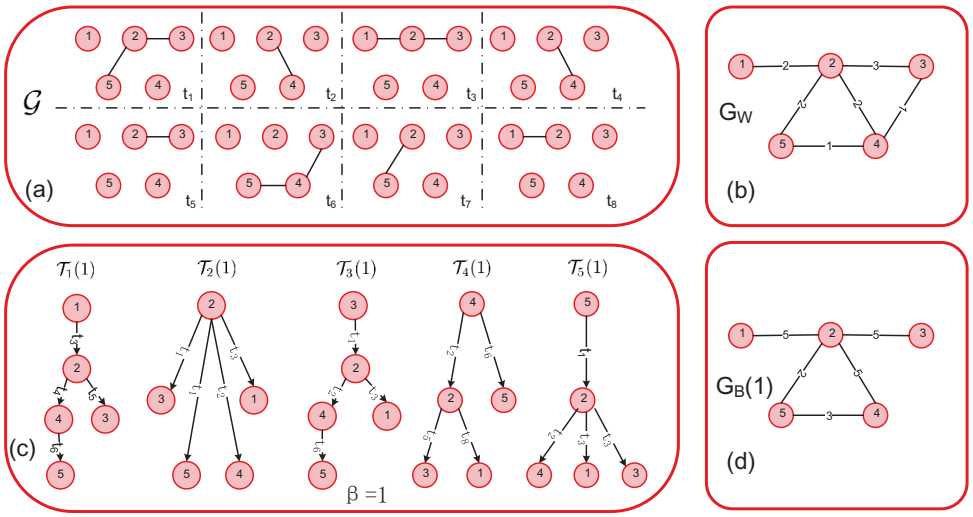


Figure 2.1: (a) A temporal network $\mathscr{G}$ with $N = 5$ nodes and $T = 8$ time steps. (b) The integrated weighted network G_W, in which a link exists between a node pair in G_W as long as there is at least one contact between them in $\mathscr{G}$. The weight of a link in G_W is the number of contacts between the two nodes in $\mathscr{G}$. (c) Diffusion path tree $\mathscr{T}_i(\beta)$, where node i is the seed and infection rate is $\beta = 1$. (d) Diffusion backbone $G_B(1)$, where the infection probability $\beta = 1$ in the SI diffusion process. The weight on the node pair represents the number of times it appears in all the diffusion path trees.

2.2.2. INFORMATION DIFFUSION BACKBONE

We propose to characterize how node pairs are involved in diffusion processes by constructing information diffusion backbones. We will construct a backbone for the SI diffusion process with a given infection probability β on a temporal network defined above. We start with the simplest case when $\beta = 1$. At time step $t = 0$, the seed node i is infected and all the other nodes are susceptible. The trajectory of the SI diffusion on $\mathscr{G}$ can be recorded by a *diffusion*

path tree $\mathcal{T}_i(\beta)$. The diffusion path tree $\mathcal{T}_i(\beta)$ records the union of contacts, via which information diffuses. We define the diffusion backbone $G_B(\beta) = (\mathcal{N}, \mathcal{L}_B(\beta))$ as the union of all diffusion path trees, i.e., $\bigcup_{i=1}^{N} \mathcal{T}_i(\beta)$, that start at each node as the seed node. The node set of $G_B(\beta)$ is $\mathcal{N}$, and nodes are connected in $G_B(\beta)$ if they are connected in any diffusion path tree. Each link in $\mathcal{L}_B(\beta)$ is associated with a weight w^B_{jk}, which denotes the number of times node pair (j,k) appears in all diffusion path trees. An example of how to construct the diffusion backbone is given in Figure 2.1(c) and (d) for $\beta = 1$. The ratio $\frac{w^B_{jk}}{N}$ indicates the probability that the node pair (j,k) appears in a diffusion trajectory starting from an arbitrary seed node.

When $0 < \beta < 1$, the diffusion process is stochastic. In this case, the backbone can be obtained as the average of a number of realizations of the backbones. Per realization, we run the SI process starting from each node serving as the seed for information diffusion, obtain the diffusion path trees and construct one realization of the diffusion backbone. The weight w^B_{jk} of a link in $G_B(\beta)$ is the average weight of this link over the h realizations. The computational complexity of constructing $G_B(\beta)$ is $\mathcal{O}(N^3 T h)$, where T is the length of the observation time window of the temporal network.

2.2.3. EMPIRICAL NETWORKS

DESCRIPTION AND BASIC FEATURES

For the construction and analysis of diffusion backbones, we consider a large number of temporal networks that capture two types of contacts, i.e., physical and virtual contacts. We collect the data sets *Reality mining* [24, 25], *Hypertext 2009* [26, 27], *High School 2011* [28], *High School 2012* [28], *High School 2013* [29], *Primary School* [30], *Workplace* [31], *Haggle* [32, 33] and *Infectious* [34] that record the face-to-face physical contacts of individuals at MIT, ACM Hypertext 2009 conference, a high school, a primary school, a workplace and the Science Gallery, respectively. We also consider virtual contact datasets recording the mailing and message behavior, including *Manufacturing Email* [35, 36], *Email Eu* [37], *DNC Email* [38] and *Collegemsg* [39]. The list of the datasets used and their detailed statistics are given in Table 2.1. We consider only the temporal network topologies measured at discrete time steps in these datasets, whereas the duration of a time step differs among these datasets. We have removed the time steps without any contact in order to consider the steps that are relevant for information diffusion and to avoid the periods that have no contact due to technical errors in measurements.

OBSERVATION TIME WINDOWS

We aim to understand which node pair is likely to be connected in the backbone, thus to contribute to a diffusion process and how such connection in the backbone is related to this node pair's temporal connection features. However, real-world temporal networks are measured for different lengths T of time windows as shown in Table 2.1. If a diffusion process has a relatively high spreading probability or the temporal network has a relatively long observation time window, almost all the nodes can be reached within a short time. The temporal contacts happened afterwards will not contribute to the diffusion process. Hence, we will select the time windows such that all contacts within each selected time window could possibly contribute, or equivalently, are relevant to a diffusion process. On the other hand, we will consider several time windows for each measured temporal network. This will allow us to

Table 2.1: Basic features of the empirical networks. The number of nodes (N), the original length of the observation time window (T in number of steps), the total number of contacts ($|\mathscr{C}|$), the number of links in G_W ($|\mathscr{L}_W|$) and contact type are shown.

Network	N	T	$\lvert\mathscr{C}\rvert$	$\lvert\mathscr{L}_W\rvert$	$Contact\ Type$
Reality Mining (RM)	96	33,452	1,086,404	2,539	Physical
Hypertext 2009 (HT2009)	113	5,246	20,818	2,196	Physical
High School 2011 (HS2011)	126	5,609	28,561	1,710	Physical
High School 2012 (HS2012)	180	11,273	45,047	2,220	Physical
High School 2013 (HS2013)	327	7,375	188,508	5,818	Physical
Primary School (PS)	242	3,100	125,773	8,317	Physical
Workplace (WP)	92	7,104	9,827	755	Physical
Manufacturing Email (ME)	167	57,791	82,876	3,250	Virtual
Email Eu (EEU)	986	207,880	332,334	16,064	Virtual
Haggle	274	15,662	28,244	2,124	Physical
Infectious	410	1,392	17,298	2,765	Physical
DNC Email (DNC)	1866	1,8682	37,421	4,384	Virtual
Collegemsg	1899	5,8911	59,835	13,838	Virtual

understand how the time window of a temporal network may influence the relation between the backbones of different spreading probabilities and relation between a node pair's local connection features and its connection in a backbone. We select the observation time windows for each measured temporal network within its original time window $[0, T]$ as follows. On each measured temporal network with its original observation time window $[0, T]$, we conduct the SI diffusion process with $\beta = 1$ by setting each node as the seed of the information diffusion process and plot the average prevalence ρ at each time step, as illustrated in Figure 2.2. The time steps are normalized by the original length of observation window T. The average prevalence at the end of the observation $t/T = 1$ is recorded as $\rho(t = T)$. The time to reach the steady state varies significantly across the temporal networks. For networks like *RM*, *HT2009*, the diffusion finishes or stops earlier and contacts happened afterwards are not relevant for the diffusion process. However, the prevalence curves ρ of the last four networks (i.e., *Haggle*, *Infectious*, *DNC* and *Collegemsg*) increase slowly and continuously over the whole period. Actually, we observe these four networks are more heterogeneous than the other networks in terms of the degree distribution of the integrated static network, which are shown in Figure 2.3.

For each real-world temporal network with its original length of observation time window T, we consider the following lengths of observation time windows: the time $T_{p\%}$ when the average prevalence reaches $p\%$, where $p \in \{10, 20, \ldots, 90\}$ and $p\% < \rho(t = T)$. For a given measured temporal network $\mathscr{G} = (\mathscr{N}, \mathscr{L})$, we consider maximally 9 observation time windows. For each length $T_{p\%}$, we construct a sub-temporal network, $\mathscr{G}_{p\%} = (\mathscr{N}, \mathscr{L}_{p\%})$, in which $\mathscr{L}_{p\%}$ includes contacts in $\mathscr{L}$ that occur earlier than $T_{p\%}$. The lengths of observation time window $T_{p\%}$ for the empirical networks are shown in Table S1 in the **APPENDIX A**. For a network like *RM*, we can get 9 sub-networks and for network like *Infectious*, we can only obtain 5 sub-networks. In total, 106 sub-networks are obtained. Contacts in all these sub-networks are relevant for SI diffusion processes with any spreading probability β. Without loss of generality, we will consider all these sub-networks with diverse lengths of observation time windows and temporal network features to study the relationship between

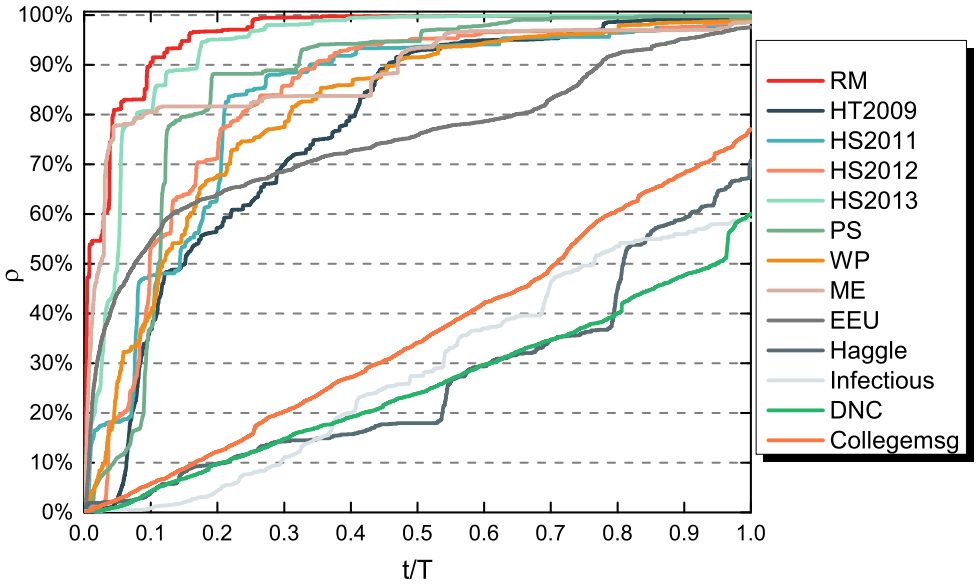


Figure 2.2: Average prevalence ρ of the SI spreading process with $\beta = 1$ on each original empirical temporal network over time. The time steps are normalized by the corresponding observation time window T of each network.

diffusion backbones and temporal connection features.

2.3. RESULTS

2.3.1. RELATIONSHIP BETWEEN DIFFUSION BACKBONES

We explore the relationships among the backbones $G_B(\beta)$ with different spreading probabilities $\beta \in [0,1]$ on the same temporal network. When the infection probability $\beta \to 0$, the backbone $G_B(\beta \to 0)$ approaches the integrated weighted network G_W if the network is finite regarding to its size and number of contacts. This can be understood as follows. When an arbitrary node i is the seed node, the probability that the information diffuses to any other node j within a given observation time window of length T is $1-(1-\beta)^{w_{ij}} = 1-e^{w_{ij}\log(1-\beta)} \sim 1-e^{-w_{ij}\beta} \sim w_{ij}\beta$, where w_{ij} is the number of contacts between the i and j within the observation time window. Assume that i and j have contact(s), i.e., $w_{ij} > 0$, and node k has no contact with the seed i but has contact(s) with node j. The probability that the information initiated by the seed i diffuses further from j to k is smaller than $w_{ij}w_{jk}\beta^2 \ll w_{ij}\beta$. In other words, the probability that the information diffuses via a second hop node pair (j,k) relative to the seed i (from the view of the integrated network) is negligibly small compared to the first hop node pair (i,j). Hence, the information diffusion tree approaches a tree whose root is the seed node and the leaves are the nodes that have contacts with the seed. The information diffusion backbone, which is the union of the diffusion trees rooted at each node, has the same topology as the integrated network. The weight w^B_{ij} of each link in the backbone is $w^B_{ij} \sim 2w_{ij}\beta$. When the network is infinite in size or number of contacts, $G_B(\beta \to 0) \sim G_W$ is not necessarily true also because a node pair can be a second hop pair relative to many seed nodes.

We denote $G_B(\beta = 0) \triangleq G_B(\beta \to 0) = G_W$ except that the weight of each node pair in the two networks is scaled. When the infection probability β is small, node pairs with more contacts are more likely to appear in the backbone. The backbone $G_B(\beta)$ varies from $G_B(0) = G_W$ when $\beta \to 0$ to $G_B(1)$ when $\beta = 1$.

OVERLAP IN LINKS BETWEEN BACKBONES

We investigate first how different these backbones with different spreading probabilities $\beta \in [0,1]$ are and whether $G_B(\beta)$ with a small and large β can be well approximated by G_W and $G_B(1)$ respectively.

However, note that the observed topology of $G_B(\beta)$ obtained from the simulation which is composed of a limited number of iterations of the spreading process can be a sub-graph of the topology of G_W. We illustrate how the number of iterations affects the ratio of links in the observed $G_B(\beta)$ to $|\mathscr{L}_W|$ in Figure S1(d-f) in the **APPENDIX B**. It shows that with the increased number h of iterations, $|\mathscr{L}_B(\beta)|$ is getting close to $|\mathscr{L}_W|$ for networks with a large observation time window. For networks a with small observation time window like $RM\_T_{10\%}$, $|\mathscr{L}_B(\beta)|$ tends to approach $|\mathscr{L}_W|$ at a small number h of iterations. For $G_B(1)$, we have $|\mathscr{L}_B(1)| <= |\mathscr{L}_W|$, which is reflected in Figure 2.5 (a) where the number of links in $G_B(0)$ and $G_B(1)$ are compared.

The similarity between two backbones or two weighted networks in general can be measured by their overlap in links or node pairs with a high weight. For each backbone $G_B(\beta)$, links in $\mathscr{L}_B(\beta)$ are ordered according to their weights in the backbone in a descending order . Thus the links in the relatively top positions are more likely to be used in the diffusion process. Therefore, for any backbone with $\beta \in [0,1]$, we consider the top $|\mathscr{L}_B(1)|$ links from $\mathscr{L}_B(\beta)$, which are denoted as $\mathscr{L}_B^*(\beta)$. The similarity or overlap between two backbones like $G_B(\beta)$ and $G_B(\beta = 0)$ can be measured by the overlap between $\mathscr{L}_B^*(\beta)$ and $\mathscr{L}_B^*(0)$, defined as

$$r(\beta, 0) = r(\mathscr{L}_B^*(\beta), \mathscr{L}_B^*(0)) = \frac{|\mathscr{L}_B^*(\beta) \cap \mathscr{L}_B^*(0)|}{|\mathscr{L}_B^*(\beta)|}, \tag{2.2}$$

For each temporal network, we construct each backbone $G_B(\beta)$, where $\beta = 0.25, 0.5, 0.75, 1$, as the average of $h = 100$ iterations of the SI spreading processes starting from each node as the seed, based on the method illustrated in Section 2.2 (The validation that 100 iterations are enough to get a stable backbone is given in Figure S1 in the **APPENDIX B**). The backbone $G_B(\beta = 0)$ equals G_W. The overlap between backbones for dataset *RM* are shown in Figure 2.4 as an example. More examples are given in Figure S2 in the **APPENDIX C**). The overlap $r(\beta, 0)$ tends to decrease with the increase of β and $G_B(\beta = 0)$ well approximates the backbones with a small β. Similarly, $G_B(1)$ well approximates the backbones with a large β. When the observation time window of a temporal network is small, the backbones with different β are relatively similar in topology. In this case, a diffusion path tree tends to have a smaller average depth (The average depth of a tree is the average number of links in the shortest path from the root to another random node in the tree) and a node pair with a large number of contacts is likely to appear or connect in the backbone, which explains why G_W approximates all the backbones including $G_B(1)$. These observations motivate us to explore the two extreme backbones $G_B(0)$ and $G_B(1)$ regarding to how much they differ from or related to each other.

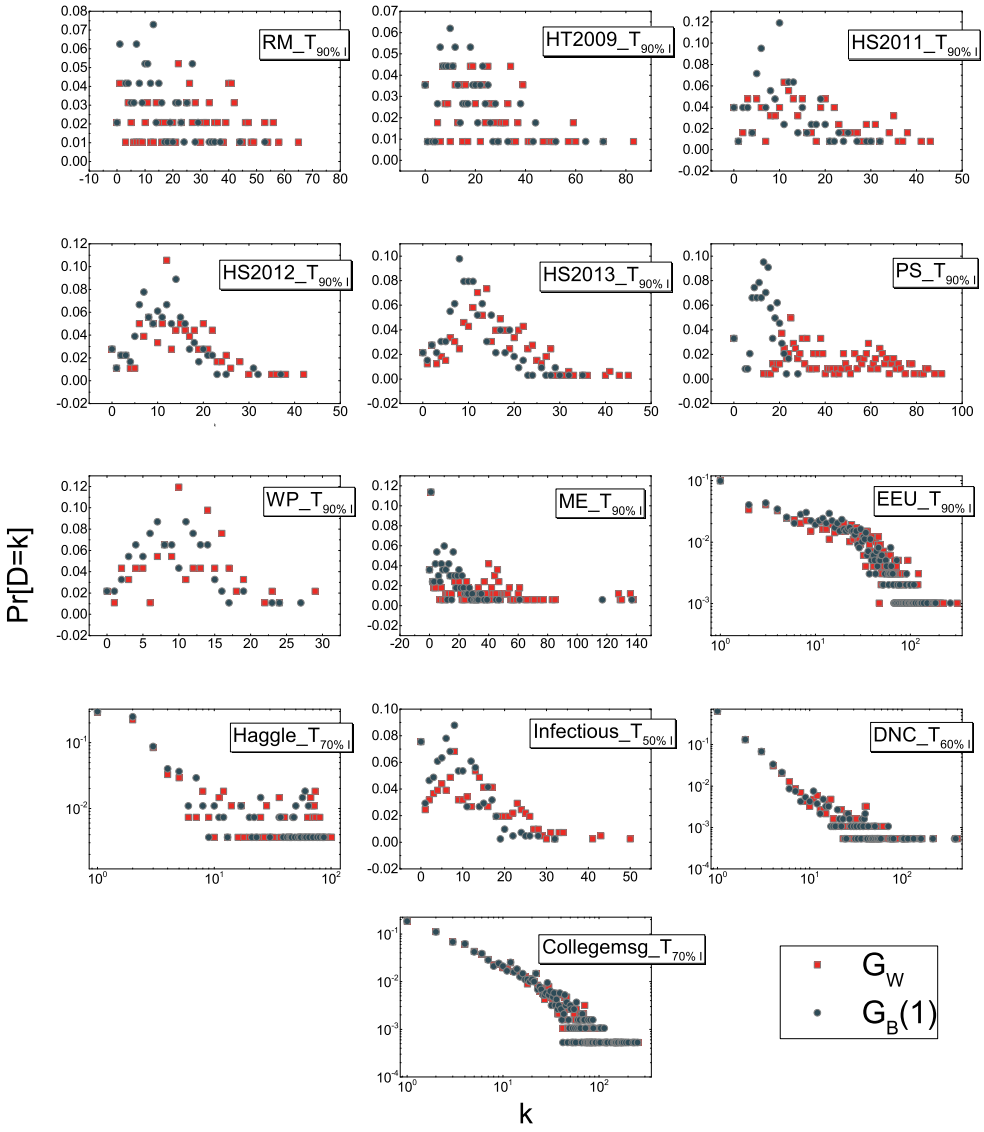


Figure 2.3: Degree distribution of G_W and $G_B(1)$ for empirical networks with longest observation window.

DEGREE OF A NODE IN DIFFERENT BACKBONES

From now on, we focus on the two extreme backbones $G_B(0) = G_W$ and $G_B(1)$. A node pair that has contact(s) may not necessarily contribute to a diffusion process. Hence, the degree of a node in $G_B(0)$ is larger or equal to its degree in $G_B(1)$. The comparison of the number of links in $G_B(0)$ and $G_B(1)$ in Figure 2.5 shows that $G_B(1)$ indeed has less links than $G_B(0)$, especially when the observation time window is large. As explained earlier, $G_B(1)$ and $G_B(0)$ are similar to each other in topology when the observation time window is small.

Furthermore, we explore the degree of a node in $G_W = G_B(0)$ and $G_B(1)$ respectively. In-

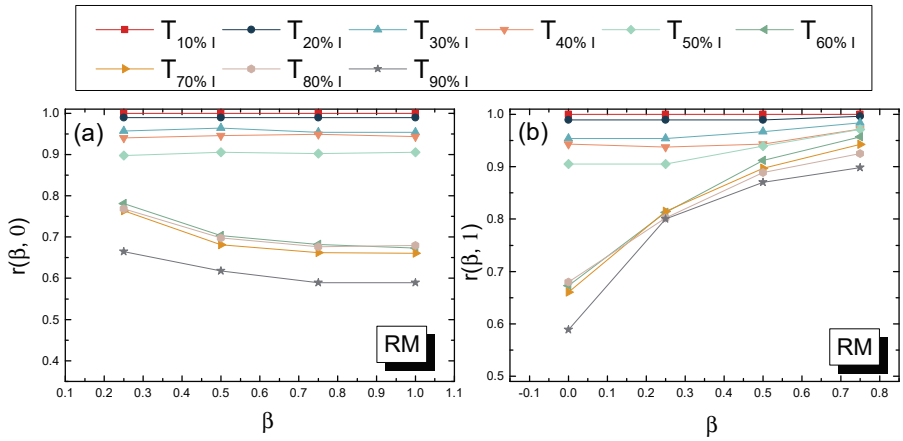


Figure 2.4: (a) Overlap $r(\beta,0)$ between $G_B(\beta)$ and $G_B(0)$ as a function of β in (sub)networks derived from dataset *RM*; (b) Overlap $r(\beta,1)$ between $G_B(\beta)$ and $G_B(1)$ as a function of β in (sub)networks derived from dataset *RM*. Diffusion backbones ($0<\beta<1$) are obtained over 100 iterations.

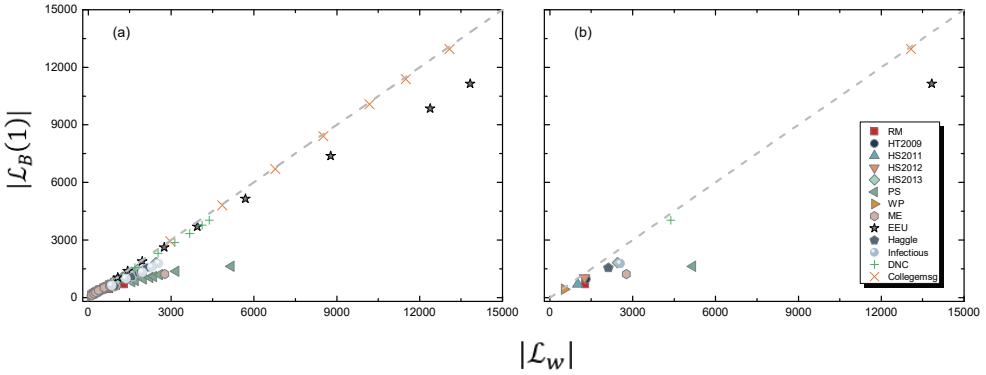


Figure 2.5: The relationship between the number of links in G_W and $G_B(1)$ for (a) all the networks with observation windows given in Table S1 in **APPENDIX A**; (b) the networks with the longest observation windows in each dataset.

terestingly, a universal finding is that the degree of a node in these two backbones tends to be linearly and positively correlated in all the empirical networks. Table S2 in the **APPENDIX E** provides the Pearson correlation coefficient between the degree of a node in G_W and in $G_B(1)$, which is above 0.7 for all the networks. Since the topology of $G_B(1)$ is a sub-graph of G_W, the degrees of a node in these two networks tend to be linearly correlated if these two networks have a similar number of links. This explains the high degree correlation when the temporal networks have a short observation window. Figure 2.6 shows the scatter plot of the degree of each node in G_W and $G_B(1)$ respectively for the network with the longest observation window when their backbones G_W and $G_B(1)$ differ much in the number of links derived from two datasets respectively. The strong degree correlation in all these cases suggests that a node with a high degree in G_W tends to have a high degree in $G_B(1)$. A node that has contacts with many others tends to be able to propagate the information directly to many others.

Is this because the degree distribution in G_W is highly heterogeneous that overrules the

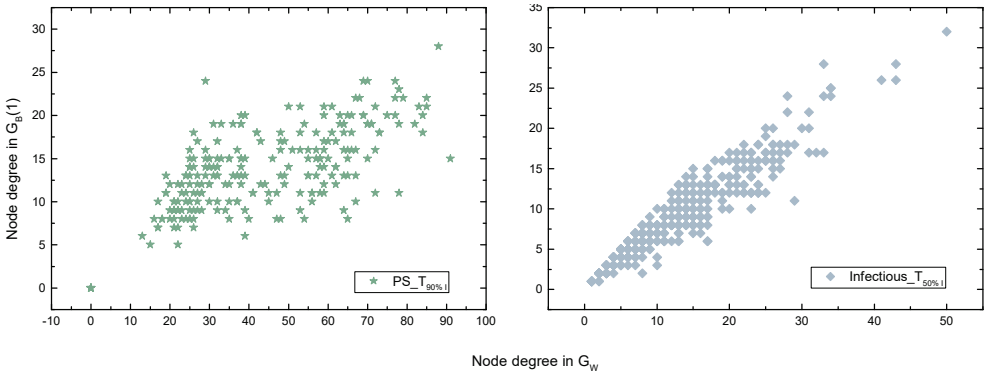


Figure 2.6: Degree correlation between G_W and $G_B(1)$ for networks *PS* and *Infectious* with the longest observation window respectively.

temporal orders of the contacts in determining how many other nodes a node is able to reach directly? Figure 2.3 shows the degree distributions in G_W and $G_B(1)$ respectively for each temporal network dataset with its longest observation window as given in Table S1 in **APPENDIX A** when these two backbones differ the most. We find that the degree distributions in these two backbones respectively indeed share a similar shape, which again support the strong linear correlation between the degrees of a node in these two backbones. However, not all networks G_W have a power-law degree distribution. The strong degree correlation between G_W and $G_B(1)$ exists even when G_W has a relatively homogeneous degree distribution. This observation motivates us to explore whether a node pair with a high degree product in G_W thus also in $G_B(1)$ tends to be connected in $G_B(1)$ in Section 2.3.1.

The degree of a node j in $G_B(1)$ tells maximally how many nodes it could propagate the information directly to given that each node is possibly the source of the information, but not necessarily how frequently this node contributes or engages in an information diffusion process when $\beta = 1$. The latter is reflected from the node strength of a node in $G_B(1)$: $\sum_{k=1}^{N} w_{jk}^{B}(\beta = 1)$.

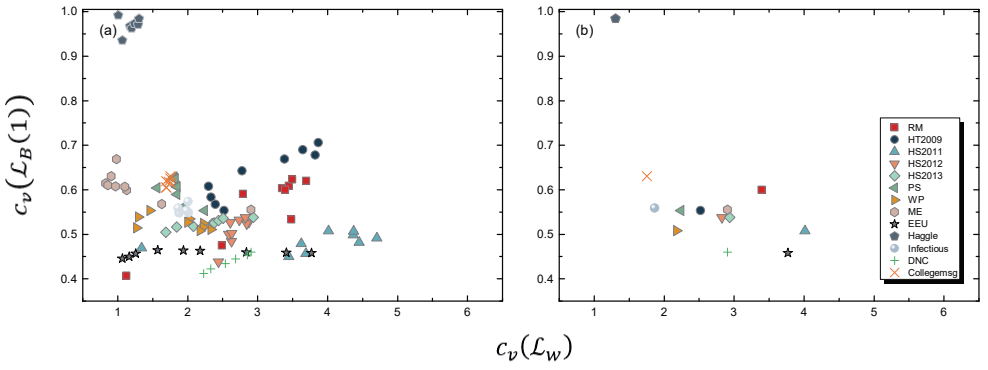


Figure 2.7: The relationship between the coefficient of variation c_v of the weight distribution in G_W and $G_B(1)$ for (a) all the networks with observation windows given in Table S1 in **APPENDIX A**; (b) all the networks with longest observation windows.

LINK WEIGHT VARIANCE IN DIFFERENT BACKBONES

The standard deviation of link weights in a backbone indicates how much the links differ in their probability of appearing in a diffusion process. We compare the standard deviation of a link weight normalized by its mean $c_v = \frac{\sqrt{\text{Var}[W^B]}}{\text{E}[W^B]}$ (which is called the coefficient of variation) in $G_B(1)$and $G_B(0)$. Figure 2.7 shows that the link weights in $G_B(0)$ or equivalently G_W are more heterogeneous than those in $G_B(1)$ for almost all the networks we considered. The relatively homogeneous link weights in $G_B(1)$ implies that predicting which node pairs tend to have a high weight in $G_B(1)$ can be challenging.

2.3.2. IDENTIFYING THE DIFFUSION BACKBONE $G_B(1)$

In this section, we investigate how to identify the (high weight) links in the backbone $G_B(1)$ based on local and temporal connection features of each node pair. The key objective to understand how a node pair's local and temporal connection features are related to its role in the global diffusion backbone $G_B(1)$. Our investigation may also allow us to approximate the backbone, whose computational complexity is high ($\mathcal{O}(N^3T)$) base on local temporal features whose computational complexity is low.

We propose to consider systematically a set of local temporal features for node pairs and examine whether node pairs having a higher value of each feature/metric tend to be connected in the backbone $G_B(1)$. Some of these features are derived from the integrated network G_W whereas the feature *Time-Scaled Weight* that we will propose encodes also the time stamps of the contacts between a node pair. These node pair features or metrics include:

• *Time-Scaled Weight* of a node pair (j,k) is defined as

$$\phi_{jk}(\alpha) = \sum_{m=1}^{n} (\frac{1}{t_{jk}^{(m)}})^{\alpha} \tag{2.3}$$

where n is the total number of contacts between j and k over the given observation window and $t_{jk}^{(i)}$ is the time stamp when the $i-th$ contact occurs and α is the scaling parameter to control the contribution of temporal information. For the node pairs that have no contact, we assume their temporal weights to be zero. This metric is motivated by the intuition that when each node is set as the seed of the diffusion process at time $t = 0$, the contacts that happen earlier have a higher probability to be used for the actual information diffusion, thus appear in $G_B(1)$. When $\alpha = 0$, $\phi_{jk}(0) = w_{jk}^B(\beta = 0)$ degenerates to the weight of the node pair in G_W. Larger α implies the node pairs with early contacts have a higher time-scaled weight.

• *Degree Product* of a node pair (j,k) refers to $d_j(\beta = 0)d_k(\beta = 0)$, the product of the degrees of j and k in the integrated network G_W. If two nodes are not connected in G_W, their degree product is zero. The motivation for this measure is as follows. Given the degree of each node in $G_B(1)$ and if the links are randomly placed, the probability that a node pair (i,j) is connected in $G_B(1)$ is proportional to $d_j(\beta = 1) \cdot d_k(\beta = 1)$. We have observed in Section 2.3.1 that the degree of a node in G_W and $G_B(1)$ are strongly and positively correlated. Moreover, only node pairs connected in G_W are possible to appear or be connected in $G_B(1)$. If the connections in $G_B(1)$ are as random as in the configuration model [40], node pairs with a high degree product $d_j(\beta = 0) \cdot d_k(\beta = 0)$ tend to appear in $G_B(1)$.

• *Strength Product* of a node pair (j,k) refers to $s_j(\beta = 0) \cdot s_k(\beta = 0)$, the product of the node strengths of j and k in the integrated network G_W, where the node strength $s_j(\beta = 0) = \sum_{i \in \mathcal{N}} A(j,i)$ of a node in G_W equals the total weight of all the links incident to this node

[41, 42]. If two nodes are not connected in G_W, their strength product is zero. This measure is an extension of the degree product to weighted networks.

• *Betweenness* of a link in G_W counts the number of shortest paths between all node pairs that traverse the link. The distance of each link, based on which the shortest path is computed, is considered to be $\frac{1}{w^B_{jk}(\beta=0)}$. In other words, inversely proportional to its link weight in G_W, since a node pair with more contacts tend to propagate information faster [43, 44]. Node pairs that are not connected in G_W have a betweenness 0. Betweenness is not local, but considered here as a benchmark feature that has been widely studied.

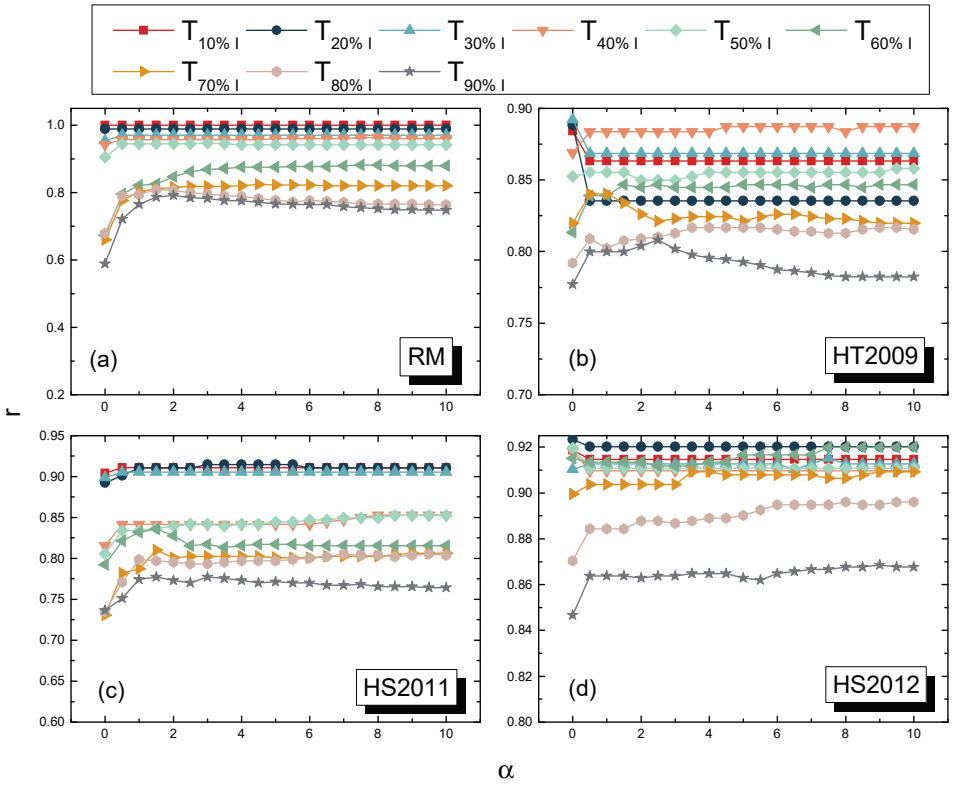


Figure 2.8: The quality of identifying links in $G_B(1)$ by using the time-scaled weight $\phi_{jk}(\alpha)$ as a function of α in temporal networks derived from datasets (a) *RM*, (b) *HT2009*, (c) *HS2011* and (d)*HS2012*.

We explore further whether these node pair features could well identify the connection of node pairs in $G_B(1)$. According to the definition of the aforementioned centrality metrics, a higher value of a metric may suggest the connection of the corresponding node pair in $G_B(1)$. According to each metric, we rank the node pairs and the $|\mathscr{L}_B(1)|$ node pairs with the highest values are identified as the links in $G_B(1)$. The identification quality of a metric, e.g., the time-scaled weight $\phi_{jk}(\alpha)$, is quantified as the overlap $r(\phi_{jk}(\alpha), 1)$ between the identified link set and the link set $\mathscr{L}_B(1)$ in $G_B(1)$, as defined by Eq. (2).

Before we compare all the metrics in their identification powers, we examine first how the scaling parameter α in the time-scaled weight $\phi_{jk}(\alpha)$ influences its identification quality. Figure 2.8 and Figure S3 in the **APPENDIX D** show that the quality differs mostly when

$0 \le \alpha \le 2$ and remains relatively stable when $\alpha \ge 2$ in all the temporal networks. Hence, we will confine ourselves to the range $0 \le \alpha \le 2$.

The quality r by using each metric versus the ratio $\frac{|\mathscr{L}_B(1)|}{|\mathscr{L}_W|}$ of the number of links in $G_B(1)$ to that in G_W are plotted in Figure 2.9 for all the empirical temporal networks, with different lengths of the observation time windows. The diagonal curve $r = \frac{|\mathscr{L}_B(1)|}{|\mathscr{L}_W|}$ corresponds to the quality of the random identification, where $|\mathscr{L}_B(1)|$ links are randomly selected from the links in G_W as the identification for the links in $G_B(1)$. Degree product, strength product and betweenness perform, in general, worse than or similarly to the random identification. Even if the connections in $G_B(1)$ were random given the degree of each node in $G_B(1)$, the quality r of identifying links in $G_B(1)$ by using the degree product is close to that of the random identification, if the distribution of the degree product is relatively homogeneous or if the $\frac{|\mathscr{L}_B(1)|}{|\mathscr{L}_W|}$ is large. The degree distribution in $G_B(1)$ is indeed relatively homogeneous and $\frac{|\mathscr{L}_B(1)|}{|\mathscr{L}_W|}$ is large in most empirical networks. This explains why the degree product performs similarly to the random identification.

The link weight in G_W, equivalently, $\phi_{jk}(\alpha = 0)$, outperforms the random identification, whereas the time-scaled weight $\phi_{jk}(\alpha)$ with a larger α performs better. Node pairs with many contacts that occur early in time tend to contribute to the actual information propagation, i.e., be connected in $G_B(1)$. This observation suggests that the temporal information is essential in determining the role of nodes in a spreading process.

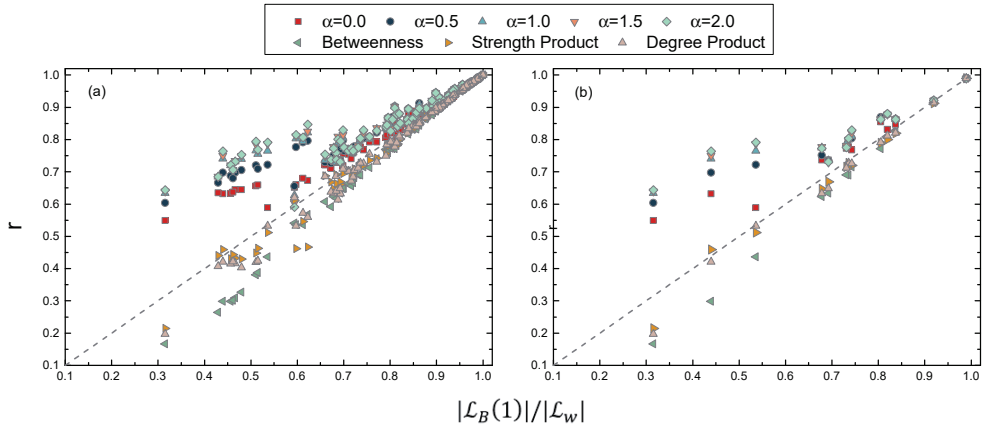


Figure 2.9: The quality of identifying links in $G_B(1)$ by using each metric for (a) all the networks with observation windows given in Table S1 in **APPENDIX A**; (b) all the networks with longest observation windows. The time-scaled weight with different α values are considered.

We investigate also whether these metrics can identify the links with the highest weights in $G_B(1)$. The quality r, as defined earlier, of identifying the top f fraction of links with the highest weight in $G_B(1)$ is plotted in Figure 2.10. We choose the top $f|\mathscr{L}_B(1)|$ node pairs according to each metric as the identification of the top $f|\mathscr{L}_B(1)|$ links in $G_B(1)$ with the highest weights. We consider the networks with the longest observation window from each dataset. The diagonal curve $r = f\frac{|\mathscr{L}_B(1)|}{|\mathscr{L}_W|}$ corresponds to the quality of the random identification. Similar to the identification of all the links in $G_B(1)$, the time-scaled weight $\phi_{jk}(\alpha)$ with a large α performs the best in identifying highly weighted links in $G_B(1)$, emphasizing again the important role of the temporal information of contacts.

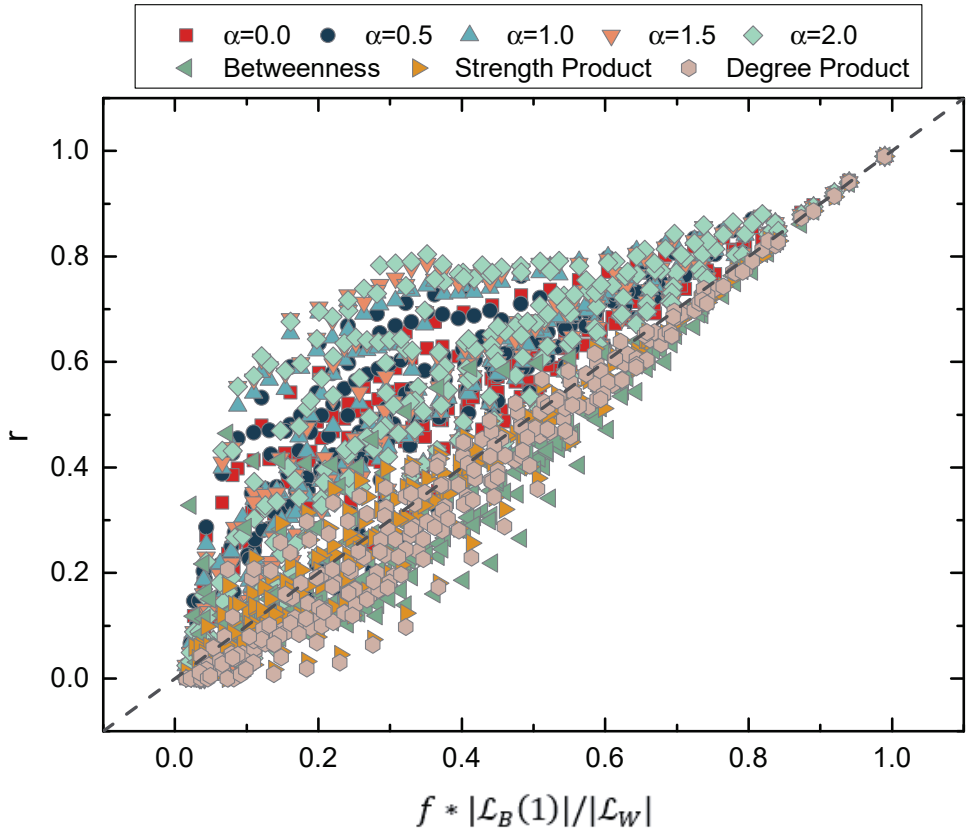


Figure 2.10: The quality r of identifying top weighted links in $G_B(1)$ by using each metric for all the networks with longest observation windows in each dataset. The time-scaled weight with different α values are considered.

2.4. CONCLUSION

Much effort has been devoted to understand how temporal network features influence the prevalence of a diffusion process. In this work, we addressed the further question: node pairs with what kind of local and temporal connection features tend to appear in a diffusion trajectory or path, thus contribute to the actual information diffusion? We consider the Susceptible-Infected spreading process with an infection probability β per contact on a temporal network as the starting point. We illustrate how to construct the information diffusion backbone $G_B(\beta)$ where the weight of each link tells the probability that a node pair appears in a diffusion process starting from a random node. We unravel how these backbones corresponding to different infection probabilities relate to each other with respect to their topology (overlap in links), the heterogeneity of the link weight, and the correlation in node degree. These relations point out the importance of two extreme backbones: $G_B(1)$ and the integrated network $G_B(0) = G_W$, between which $G_B(\beta)$ varies. We find that the temporal node pair feature that we proposed could better identify the links in $G_B(1)$ as well as the high weight links than the features derived from the integrated network. This universal finding across all the empirical networks highlights that temporal information are crucial in determining a node pair's role in a diffusion process. A node pair with many early contacts tends to appear in a diffusion process. We have also used rank correlation like Kendall and Spearman to evaluate the quality of time-scaled weight in identifying the precise weight ranking of all the links in $G_B(1)$. However, we found that the time-scaled weight when $\alpha = 0$ performs the best, which means the temporal node pair feature is not ideal to identify the exact importance of the links in the backbone $G_B(1)$. Therefore, how to predict the ranking of the link weights in the backbone remains as an interesting future question.

This work reminds us the studies a decade ago about the information transportation via the shortest path on a static network [44]. How frequently a link appears in a shortest path thus contributes to the transportation of information is reflected by the weight of the link in the backbone or overlay, the union of shortest paths between all node pairs [45]. This weight equals the betweenness, which has a high computational complexity, thus motivated the exploration how a node pair's local connection features are related to its betweenness.

The study of information diffusion paths on a temporal network is more complex due to the extra dimension of time. Our finding that early contacts with a quadratic decay in weight over time indicates the appearance of a node pair in a diffusion path, suggests the possibility to identify the appearance of a node pair in a diffusion path in a long period based on its early contacts within a short period, an interesting follow-up question. This work opens new challenging questions like which nodes tend to be reached early and more likely by the information, how such heterogeneous features at node or link level are related to local temporal connection features. In addition, other spreading models like social contagions and coevolution spreading models can be further considered beyond the SI spreading model studied here [1, 2, 46–49]. Our findings may inspire the exploration of optimization problems such as which node pairs or contacts should be stimulated (e.g. added) in order to maximize the prevalence of an information diffusion process. Stimulating early contacts seems essential but adding them between which node pairs and when is non-trivial.

2.5. APPENDIX

A: DATA DESCRIPTION

Table 2.2: The lengths of the observation time window that we choose based on the average prevalence ρ when $\beta = 1$. For instance, $T_{90\%}$ represents the time when the prevalence reaches $\rho = 90\%$.

Network	$T_{90\%}$	$T_{80\%}$	$T_{70\%}$	$T_{60\%}$	$T_{50\%}$	$T_{40\%}$	$T_{30\%}$	$T_{20\%}$	$T_{10\%}$
RM	3325	1482	1278	987	257	133	111	34	5
HT2009	2394	2131	1575	1154	790	568	439	377	332
HS2011	1903	1177	1152	1001	805	447	425	396	47
HS2012	3915	2680	1907	1481	1109	1043	925	675	403
HS2013	1253	583	406	395	369	236	195	113	50
PS	997	510	378	359	347	323	287	276	136
WP	3328	2186	1538	1133	832	708	400	320	218
ME	27189	5096	1885	1735	1387	731	461	285	168
EEU	160710	134342	67883	27531	15792	8100	4047	2348	1490
Haggle	/	/	15640	14229	12668	12440	9523	8416	3293
Infectious	/	/	/	/	1062	955	751	553	410
DNC	/	/	/	18680	17712	14918	11420	7817	3860
Collegemsg	/	/	54493	46419	41663	33889	26018	17367	9747

B: NUMBER OF ITERATIONS TO COMPUTE THE BACKBONE

We explore whether $h = 100$ iterations is sufficient to get a representative backbone when $0 < \beta < 1$. Given the temporal network and β, we first construct the diffusion backbones by choosing the number of iterations as $50, 100, 200, 300, 400, 500$, and then we compute the overlap r between the backbone obtained as the average of 100 iterations with the backbones obtained as the average of $50, 200, 300, 400, 500$ iterations, respectively. The overlap r is defined the same as Eq. (2). As the complexity of computing backbones is high, we consider a large number of networks but not all. Figure 2.11(d-f) shows the ratio of links in the observed $G_B(\beta)$ to $|\mathscr{L}_W|$ slightly increases with the increase of h. The overlap r is in general high, above 0.95 (Figure 2.11(a-c)). These observations support that we could obtain a relatively representative backbone as the average of 100 realizations of the backbone constructions. In addition, the slightly increase of link ratio also supports that the observed topology of $G_B(\beta)(0 < \beta < 1)$ is approaching G_W when the iteration times h is large enough.

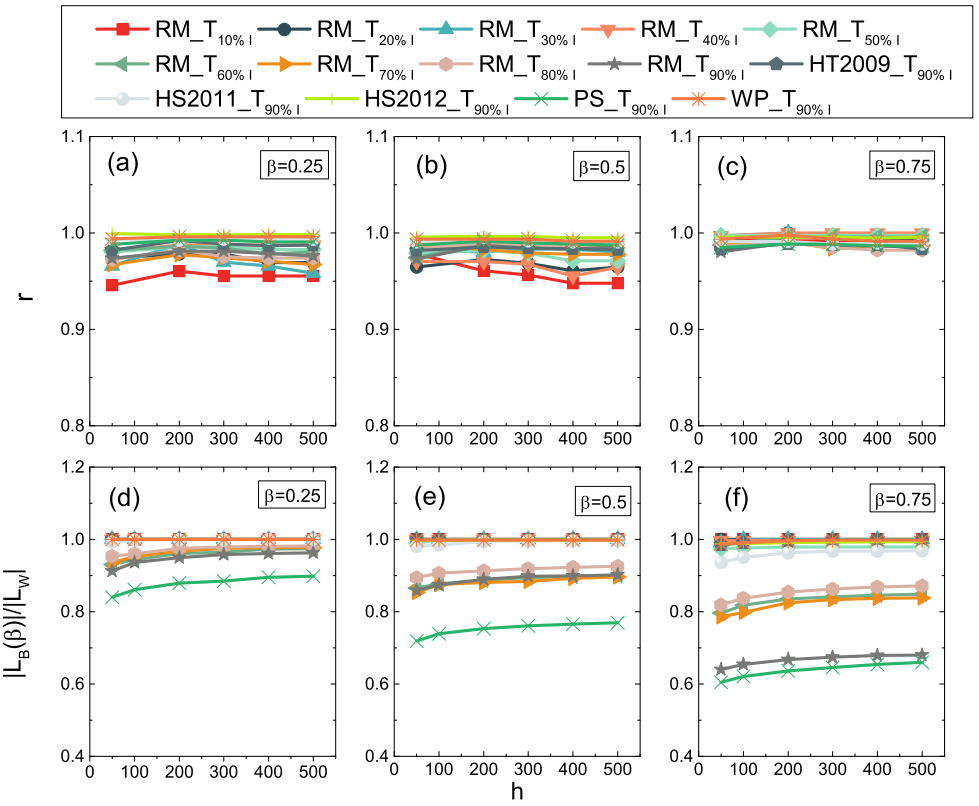


Figure 2.11: (a-c) Overlap r between backbone obtained from 100 iterations with the backbones obtained from $h = 50, 200, 300, 400, 500$ iterations on different temporal networks. (d-f) The ratio of links in the observed $G_B(\beta)$ to $|\mathscr{L}_W|$ in the backbones as a function of the number of iterations.

C: RELATIONSHIP BETWEEN BACKBONES

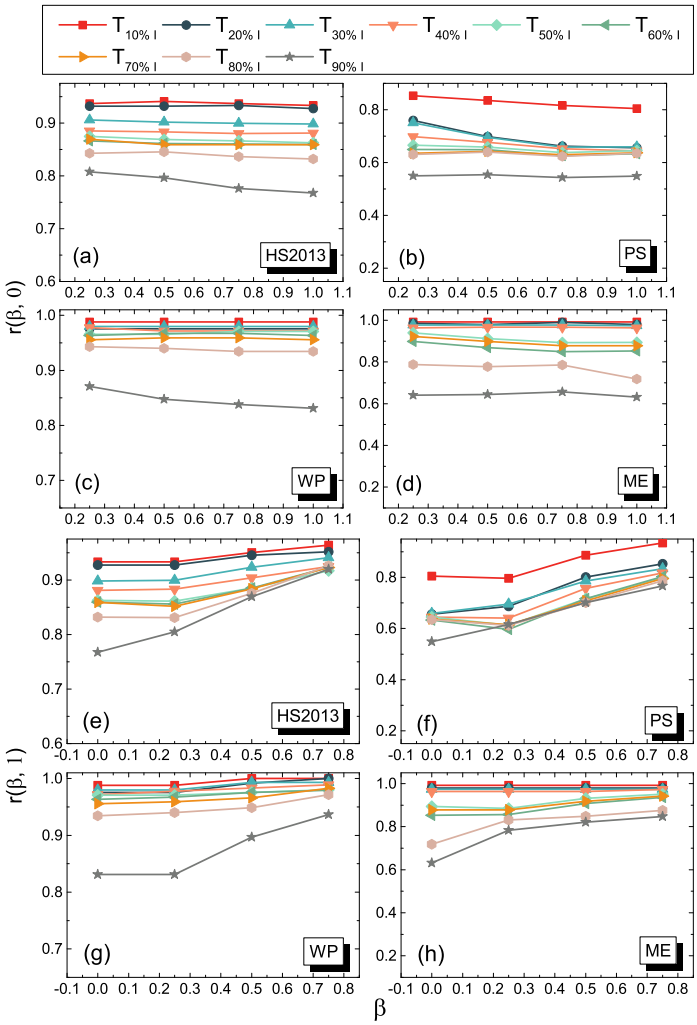


Figure 2.12: (a-d) Overlap $r(\beta,0)$ between $G_B(\beta)$ and $G_B(0)$ as a function of β in (sub)networks derived from dataset *HS2013*, *PS*, *WP* and *ME*; (e-h) Overlap $r(\beta,1)$ between $G_B(\beta)$ and $G_B(1)$ as a function of β in (sub)networks derived from dataset *HS2013*, *PS*, *WP* and *ME*. Diffusion backbones ($0<\beta<1$) are obtained from 100 iterations.

D: IDENTIFICATION OF LINKS IN $G_B(1)$

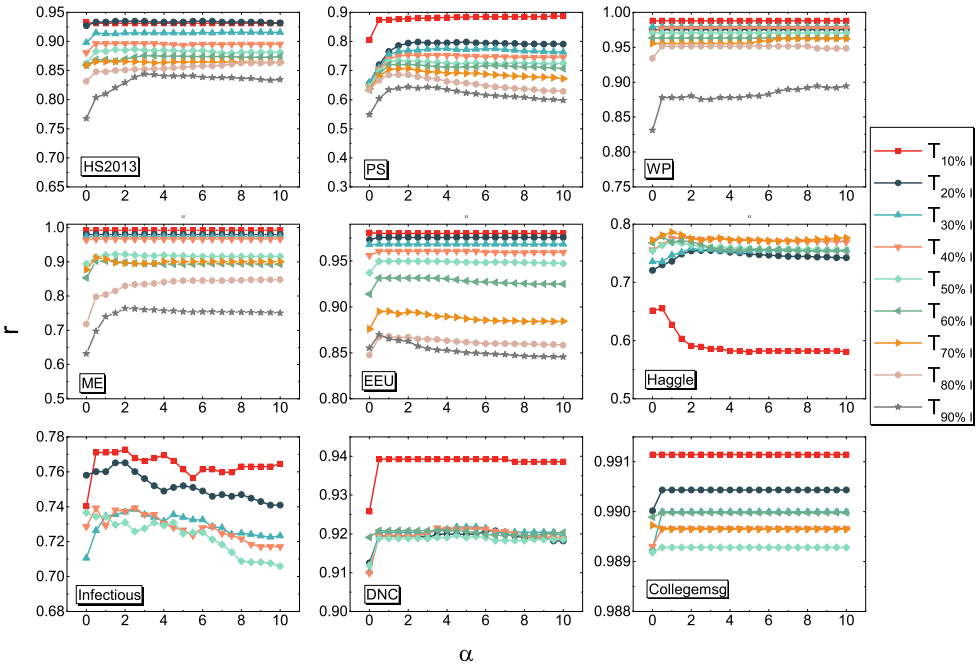


Figure 2.13: The quality of identifying links in $G_B(1)$ by using the time-scaled weight $\phi_{jk}(\alpha)$ as a function of α in temporal networks derived from datasets (a) *HS2013*; (b) *PS*; (c) *WP*; (d) *ME*; (e)*EEU*; (f) *Haggle*; (g) *Infectious*; (h) *DNC*; (i) *Collegemsg*.

E: DEGREE CORRELATION BETWEEN G_W AND $G_B(1)$

Table 2.3: Pearson correlation coefficient $P(G_W, G_B(1))$ between node degree in G_W and $G_B(1)$ in all the networks.

Network	$T_{90\%}$	$T_{80\%}$	$T_{70\%}$	$T_{60\%}$	$T_{50\%}$	$T_{40\%}$	$T_{30\%}$	$T_{20\%}$	$T_{10\%}$
RM	0.85	0.87	0.84	0.85	0.98	0.99	0.99	1.0	1.0
HT2009	0.97	0.97	0.98	0.98	0.99	0.99	1.0	0.99	0.98
HS2011	0.94	0.93	0.93	0.97	0.96	0.95	0.97	0.97	0.97
HS2012	0.97	0.98	0.98	0.99	0.99	0.99	0.98	0.99	0.99
HS2013	0.94	0.96	0.97	0.97	0.97	0.98	0.98	0.99	0.99
PS	0.70	0.76	0.78	0.80	0.82	0.82	0.80	0.78	0.91
WP	0.94	0.99	0.99	0.99	0.99	0.99	1.0	0.99	1.0
ME	0.76	0.94	0.99	0.93	0.99	1.0	1.0	1.0	1.0
EEU	0.99	0.99	0.99	0.99	0.99	1.0	1.0	1.0	1.0
Haggle	/	/	0.99	0.99	0.98	0.98	0.98	0.98	0.97
Infectious	/	/	/	/	0.94	0.94	0.93	0.93	0.93
DNC	/	/	/	1.0	1.0	1.0	0.99	0.99	0.99
Collegemsg	/	/	1.0	1.0	1.0	1.0	1.0	1.0	1.0

REFERENCES

[1] D. J. Watts, *A simple model of global cascades on random networks,* Proceedings of the National Academy of Sciences **99**, 5766 (2002).

[2] M. Granovetter, *Threshold models of collective behavior,* American journal of sociology **83**, 1420 (1978).

[3] R. Pastor-Satorras, C. Castellano, P. Van Mieghem, and A. Vespignani, *Epidemic processes in complex networks,* Reviews of modern physics **87**, 925 (2015).

[4] C. Liu, X.-X. Zhan, Z.-K. Zhang, G.-Q. Sun, and P. M. Hui, *How events determine spreading patterns: information transmission via internal and external influences on social networks,* New Journal of Physics **17**, 113045 (2015).

[5] Z.-K. Zhang, C. Liu, X.-X. Zhan, X. Lu, C.-X. Zhang, and Y.-C. Zhang, *Dynamics of information diffusion and its applications on complex networks,* Physics reports **651**, 1 (2016).

[6] B. Qu and H. Wang, *Sis epidemic spreading with heterogeneous infection rates,* IEEE Transactions on Network Science and Engineering **4**, 177 (2017).

[7] A.-L. Barabási, *Network science* (Cambridge university press, 2016).

[8] P. Holme and J. Saramäki, *Temporal networks,* Physics reports **519**, 97 (2012).

[9] P. Holme, *Modern temporal network theory: a colloquium,* The European Physical Journal B **88**, 234 (2015).

[10] I. Scholtes, N. Wider, R. Pfitzner, A. Garas, C. J. Tessone, and F. Schweitzer, *Causality-driven slow-down and speed-up of diffusion in non-markovian temporal networks,* Nature communications **5**, 5024 (2014).

[11] E. Valdano, L. Ferreri, C. Poletto, and V. Colizza, *Analytical computation of the epidemic threshold on temporal networks,* Physical Review X **5**, 021005 (2015).

[12] Y.-Q. Zhang, X. Li, and A. V. Vasilakos, *Spectral analysis of epidemic thresholds of temporal networks,* IEEE transactions on cybernetics (2017).

[13] M. Karsai, M. Kivelä, R. K. Pan, K. Kaski, J. Kertész, A.-L. Barabási, and J. Saramäki, *Small but slow world: How network topology and burstiness slow down spreading,* Physical review E **83**, 025102 (2011).

[14] R. Lambiotte, L. Tabourier, and J.-C. Delvenne, *Burstiness and spreading on temporal networks,* The European Physical Journal B **86**, 320 (2013).

[15] A. Moinet, M. Starnini, and R. Pastor-Satorras, *Burstiness and aging in social temporal networks,* Physical review letters **114**, 108701 (2015).

[16] H. W. Hethcote, *The mathematics of infectious diseases,* SIAM review **42**, 599 (2000).

[17] L. E. Rocha and V. D. Blondel, *Bursts of vertex activation and epidemics in evolving networks,* PLoS computational biology **9**, e1002974 (2013).

[18] S. Lee, L. E. Rocha, F. Liljeros, and P. Holme, *Exploiting temporal network structures of human interaction to effectively immunize populations,* PloS One **7**, e36439 (2012).

[19] M. Starnini, A. Machens, C. Cattuto, A. Barrat, and R. Pastor-Satorras, *Immunization strategies for epidemic processes in time-varying contact networks,* Journal of theoretical biology **337**, 89 (2013).

[20] Z. Yang and T. Zhou, *Epidemic spreading in weighted networks: an edge-based mean-field solution,* Physical review E **85**, 056106 (2012).

[21] X. Chu, J. Guan, Z. Zhang, and S. Zhou, *Epidemic spreading in weighted scale-free networks with community structure,* Journal of Statistical Mechanics: Theory and Experiment **2009**, P07043 (2009).

[22] R. Pfitzner, I. Scholtes, A. Garas, C. J. Tessone, and F. Schweitzer, *Betweenness preference: Quantifying correlations in the topological dynamics of temporal networks,* Physical review letters **110**, 198701 (2013).

[23] X. Li and X. Li, *Reconstruction of stochastic temporal networks through diffusive arrival times,* Nature communications **8**, 15729 (2017).

[24] *Reality mining network dataset – KONECT,* (2017).

[25] N. Eagle and A. (Sandy) Pentland, *Reality Mining: Sensing complex social systems,* Pers. Ubiquitous Comput **10**, 255 (2006).

[26] *Hypertext 2009 network dataset – KONECT,* (2017).

[27] L. Isella, J. Stehlé, A. Barrat, C. Cattuto, J.-F. Pinton, and W. Van den Broeck, *What's in a crowd? analysis of face-to-face behavioral networks,* Journal of theoretical biology **271**, 166 (2011).

[28] J. Fournet and A. Barrat, *Contact patterns among high school students,* PloS One **9**, e107878 (2014).

[29] R. Mastrandrea, J. Fournet, and A. Barrat, *Contact patterns in a high school: a comparison between data collected using wearable sensors, contact diaries and friendship surveys,* PloS One **10**, e0136497 (2015).

[30] J. Stehlé, N. Voirin, A. Barrat, C. Cattuto, L. Isella, J.-F. Pinton, M. Quaggiotto, W. Van den Broeck, C. Régis, B. Lina, *et al.*, *High-resolution measurements of face-to-face contact patterns in a primary school,* PloS One **6**, e23176 (2011).

[31] M. Génois, C. L. Vestergaard, J. Fournet, A. Panisson, I. Bonmarin, and A. Barrat, *Data on face-to-face contacts in an office building suggest a low-cost vaccination strategy based on community linkers,* Network Science **3**, 326 (2015).

[32] *Haggle network dataset – KONECT,* (2017).

[33] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, *Impact of human mobility on opportunistic forwarding algorithms,* IEEE Transactions on Mobile Computing **6**, 606 (2007).

[34] L. Isella, J. Stehlé, A. Barrat, C. Cattuto, J.-F. Pinton, and W. Van den Broeck, *What's in a crowd? analysis of face-to-face behavioral networks,* Journal of theoretical biology **271**, 166 (2011).

[35] *Manufacturing emails network dataset – KONECT,* (2017).

[36] R. Michalski, S. Palus, and P. Kazienko, *Matching organizational structure and social network extracted from email communication,* in *Lecture Notes in Business Information Processing*, Vol. 87 (Springer Berlin Heidelberg, 2011) pp. 197–206.

[37] J. Leskovec, J. Kleinberg, and C. Faloutsos, *Graph evolution: Densification and shrinking diameters,* ACM transactions on Knowledge Discovery from Data (TKDD) **1**, 2 (2007).

[38] *Dnc emails network dataset – KONECT,* (2017).

[39] P. Panzarasa, T. Opsahl, and K. M. Carley, *Patterns and dynamics of users' behavior and interaction: Network analysis of an online community,* Journal of the American Society for Information Science and Technology **60**, 911 (2009).

[40] M. E. Newman, S. H. Strogatz, and D. J. Watts, *Random graphs with arbitrary degree distributions and their applications,* Physical review E **64**, 026118 (2001).

[41] H. Wang, L. Douw, J. M. Hernandez, J. Reijneveld, C. Stam, and P. Van Mieghem, *Effect of tumor resection on the characteristics of functional brain networks,* Physical Review E **82**, 021924 (2010).

[42] D. Grady, C. Thiemann, and D. Brockmann, *Robust classification of salient links in complex networks,* Nature communications **3**, 864 (2012).

[43] M. E. Newman, *Scientific collaboration networks. ii. shortest paths, weighted networks, and centrality,* Physical review E **64**, 016132 (2001).

[44] H. Wang, J. M. Hernandez, and P. Van Mieghem, *Betweenness centrality in a weighted network,* Physical Review E **77**, 046105 (2008).

[45] P. Van Mieghem and H. Wang, *The observable part of a network,* IEEE/ACM Transactions on Networking **17**, 93 (2008).

[46] X. Chen, W. Wang, S. Cai, H. E. Stanley, and L. A. Braunstein, *Optimal resource diffusion for suppressing disease spreading in multiplex networks,* Journal of Statistical Mechanics: Theory and Experiment **2018**, 053501 (2018).

[47] X.-X. Zhan, C. Liu, G. Zhou, Z.-K. Zhang, G.-Q. Sun, J. J. Zhu, and Z. Jin, *Coupling dynamics of epidemic spreading and information diffusion on complex networks,* Applied Mathematics and Computation **332**, 437 (2018).

[48] W. Wang, M. Cai, and M. Zheng, *Social contagions on correlated multiplex networks,* Physica A: Statistical Mechanics and its Applications **499**, 121 (2018).

[49] W. Wang, Q.-H. Liu, J. Liang, Y. Hu, and T. Zhou, *Coevolution spreading in complex networks,* Physics Reports (2019).

3

SUPPRESSING INFORMATION DIFFUSION VIA LINK BLOCKING IN TEMPORAL NETWORKS

This chapter have been published as: X.-X. Zhan, A. Hanjalic and H. Wang, Suppressing Information Diffusion via Link Blocking in Temporal Networks, In International Conference on Complex Networks and Their Applications, Springer, Cham. 448-458 (2019)

In this chapter, we explore how to effectively suppress the diffusion of (mis)information via blocking/removing the temporal contacts between selected node pairs. Information diffusion can be modelled as, e.g., an SI (Susceptible-Infected) spreading process, on a temporal social network: an infected (information possessing) node spreads the information to a susceptible node whenever a contact happens between the two nodes. Specifically, the link (node pair) blocking intervention is introduced for a given period and for a given number of links, limited by the intervention cost. We address the question: which links should be blocked in order to minimize the average prevalence over time? We propose a class of link properties (centrality metrics) based on the information diffusion backbone [1], which characterizes the contacts that actually appear in diffusion trajectories. Centrality metrics of the integrated static network have also been considered. For each centrality metric, links with the highest values are blocked for the given period. Empirical results on eight temporal network datasets show that the diffusion backbone based centrality methods outperform the other metrics whereas the betweenness of the static network, performs reasonably well especially when the prevalence grows slowly over time.

3.1. INTRODUCTION

The development of sensor technology and electronic communication service provide us access to rich human interaction data, including proximity data like human face-to-face contacting, electronic communication data like email exchange, message exchange, phone calls [2–4]. The recorded human interactions can be represented as temporal networks, in which each interaction is represented as a contact at a given time step between two nodes. The availability of such social temporal networks inspires us to explore further how to suppress the diffusion of (mis)information that unfolds on them? One possible intervention is to block the links (i.e., remove contacts between node pairs), but only for a given period and given node pairs limited by intervention cost. In this work, we address the question: which links should we block for a given period in order to minimize the prevalence averaged over time, i.e., to prevent or delay the diffusion on temporal networks?

Researchers have worked on problems on temporal networks, e.g., nodes with what temporal topological properties (temporal centrality metrics) should be selected as the seed node that starts the information diffusion in order to maximize the final prevalence [5–10], links with what temporal topological properties appear more frequently in a diffusion trajectory [1]. These works explored in general the relation between node's or link's topological properties and its role in a dynamic process on a temporal network. Our question which links should be blocked to suppress information diffusion will actually reveal the role of a link within a given period in a diffusion process in relation to the link's temporal topological properties.

As a starting point, we consider the Susceptible-Infected (SI) model as the information diffusion process. A seed node possesses the information (is infected) at time $t = 0$ whereas all the other nodes are susceptible. An infected node spreads the information to a susceptible node whenever a contact happens between the two nodes. Given a temporal network within the observation time window $[0, T]$, we would like to choose a given number of links within a period $[t_s, t_e]$ to block in order to suppress the diffusion. We propose a comprehensive set of link centrality metrics that characterize diverse temporal topological properties. Each centrality metric is used to rank the links and we remove the links with the highest centrality values for the period $[t_s, t_e]$. One group of centrality metrics is based on the information diffusion backbone [1], which characterizes how the contacts appear in an information diffusion

trajectory thus contribute to the diffusion process. Centrality metrics of the integrated static network, where two nodes are connected if they have at least one contact, are also considered. We propose as well the temporal link gravity, generalized from the static node gravity model [11]. We conduct the SI spreading on the original temporal network as well as the temporal network after link blocking. Their difference in prevalence accumulated over time is used to evaluate the performance of the link blocking strategies/metrics. Our experiments on eight real-world temporal networks show that the diffusion backbone based metrics and the betweenness of the static integrated networks evidently outperform the rest. The backbone based metrics (betweenness of static network) perform(s) better when the prevalence increases fast (slowly) over time. This observation remains universal for diverse choices of the blocking period $[t_s, t_e]$ and number of links to block. Our finding points out that both temporal and static centrality metrics, with different computational complexities, are crucial in identifying links' role in a dynamic process.

The rest of the chapter is organized as follows. We propose the methodology in Section 3.2. In Section 3.2.1, the representation of a temporal network is introduced. In Section 3.2.2, the construction of diffusion backbone is illustrated. Afterwards, we propose the link centrality metrics in Section 3.2.3. In Section 3.2.4, the link blocking procedure and the performance evaluation method are given. We further describe temporal empirical networks that will be used in Section 3.3. The results of the link blocking strategies on the temporal empirical networks are analyzed in Section 3.4. We conclude our chapter in Section 7.

3.2. METHODS

3.2.1. REPRESENTATION OF TEMPORAL NETWORKS

A temporal network within a given time window $[0, T]$ is represented as $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ denotes the node set and the number of nodes is $N = |\mathcal{N}|$. The contact set $\mathcal{L} = \{l(j,k,t), t \in [0,T], j,k \in \mathcal{N}\}$ contains the element $l(j,k,t)$ representing that a contact between node j and k occurs at time step t. The integrated weighted network of $\mathcal{G}$ is denoted by $G_W = (\mathcal{N}, \mathcal{L}_W)$. The weight w_{jk} of link $l(j,k)$ counts the number of contacts between node j and node k.

3.2.2. INFORMATION DIFFUSION BACKBONE

The information diffusion backbone was proposed to characterize how node pairs appear in a diffusion trajectory thus contribute to the actual diffusion process [1]. To illustrate our method, we construct the backbone for the SI model with infection probability $\beta = 1$, which means that an infected node infects a susceptible node with probability $\beta = 1$ whenever the two nodes have a contact. The backbone can be also constructed for the SI model with any infection probability $\beta \in [0,1]$.

We first record the spreading tree $\mathcal{T}_i$ of each node i by setting i as the seed of the SI spreading process starting at $t = 0$. The spreading tree $\mathcal{T}_i$ is the union of the contacts through which the information propagates. The diffusion backbone G_B is defined as the union of all the spreading trees, i.e., $G_B = (\mathcal{N}, \mathcal{L}_B) = \bigcup_{i=1}^{N} \mathcal{T}_i$. We use $\mathcal{N}$, $\mathcal{L}_B$ to represent the node set and the link set respectively. Each link $l(j,k)$ in $\mathcal{L}_B$ is associated with a weight w^B_{jk}, counting the number of contacts between j and k, that appear in diffusion trees/trajectories initiated from every node. An example of how we construct the diffusion backbone G_B is given in Figure 3.1(a-c).

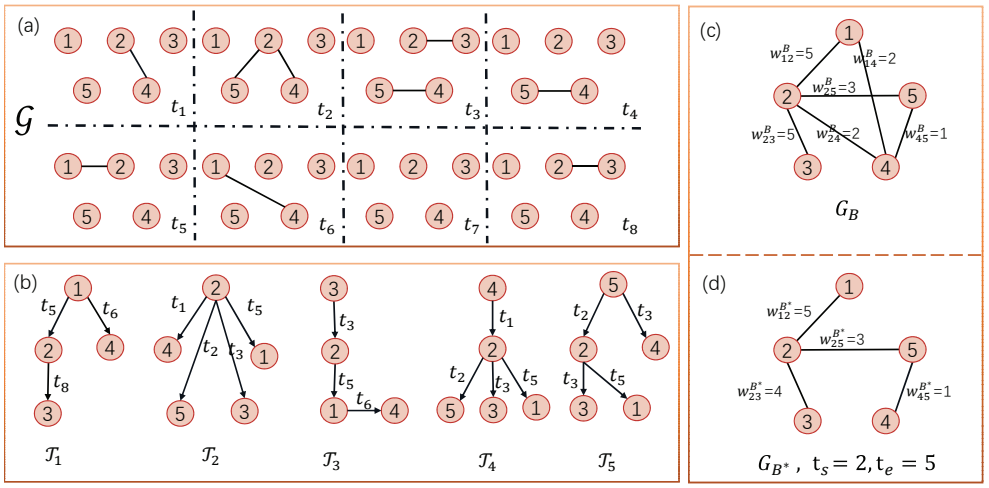


Figure 3.1: (a) A temporal network $\mathcal{G}$ with $N = 5$ nodes and $T = 8$ time steps. (b) Spreading trees rooted at every seed node. The time step on each link denotes the time of the contact through which information diffuses. (c) The diffusion backbone G_B. (d) Diffusion backbone G_{B^*} confined within $t_s = 2, t_e = 5$. When we consider the links that only appear in a time window $[t_s, t_e] = [2,5]$, the value on the link shows the link weight in G_{B^*}.

3.2.3. LINK CENTRALITY METRICS

We first propose three backbone based link centrality metrics:

• *Backbone Weight*. The backbone weight w^B_{jk} of a link $l(j,k)$ counts how many times the link or its contacts appear in spreading trees (trajectories) initialized from every node.

• *Time-confined Backbone Weight* $[t_s, t_e]$. Furthermore, we define the time-confined information diffusion backbone G_{B^*}, which generalizes our previous backbone definition. The backbone G_{B^*} confined within a time window $[t_s, t_e]$ is the union of all the spreading trees but only of the contacts that occur within $[t_s, t_e]$. Hence, two nodes in G_{B^*} are connected if at least one contact between them within $[t_s, t_e]$ appears in a diffusion tree rooted at any node. The weight $w^{B^*}_{jk}$ of link $l(j,k)$ in G_{B^*} equals to the number of times that contact(s) between j and k within $[t_s, t_e]$ that appear in the spreading trees rooted at every node. The link weight in G_{B^*} characterizes the frequency that a link, within $[t_s, t_e]$, contributes to the information diffusion. An example of the time-confined backbone construction is given in Figure 3.1(d), where $t_s = 2, t_e = 5$. Take link $l(2,4)$ as an example. It appears in the spreading trees twice, both at time step t_1, which is beyond range $[t_s = 2, t_e = 5]$. Therefore, $w^{B^*}_{24} = 0$. Link $l(2,3)$ appears at time step t_8, t_3, t_3, t_3, t_3 in all the spreading trees, only the time step t_8 is out of range [2, 5]. Hence, $w^{B^*}_{23} = 4$.

• *Backbone Betweenness*. The backbone betweenness is defined to measure the link influence in disseminating global information. Given a spreading tree $\mathcal{T}_i$, the number of descendant nodes of link $l(j,k)$ is denoted as B^i_{jk}. We define the backbone betweenness B_{jk} of link $l(j,k)$ as the average number of descendant nodes over all the spreading trees, i.e., $B_{jk} = \frac{1}{N}\sum_{i \in \mathcal{N}} B^i_{jk}$.

We consider as well the following centrality metrics derived from the integrated weighted network. Only the links in the integrated network deserves blocking. All the following metrics are zero for a node pair that they are not connected in the integrated network.

• *Degree Product* of a link $l(j,k)$ is the product of the degrees of its two end nodes in

G_W, i.e., $d_j \cdot d_k$.

• *Strength Product*. The node strength of a node j in G_W is defined as $s_j = \sum_{k \in \Gamma_j} w_{jk}$, where Γ_j is the neighbor set of node j. Hence, the strength of a node equals to the total weight of all the links incident to this node. We define strength product of a link $l(j,k)$ as $s_j \cdot s_k$.

• *Static Betweenness*. The static betweenness centrality for a link is the number of shortest paths between all node pairs that pass through the link. To compute the shortest path, we define the distance of each link in the integrated network G_W inversely proportional to its link weight in G_W. This choice follows the assumption that links with a higher weight in G_W can spread information faster [12].

• *Link Weight*. The link weight w_{jk} of a link $l(j,k)$ in G_W tells the total number of contacts between node j and k in the temporal network $\mathcal{G}$ within the observation window $[0,T]$.

• *Time-Confined Link Weight* $[t_s,\ t_e]$ refers to the number of contacts between two ending nodes that occur in $[t_s,\ t_e]$.

• *Temporal Link Gravity*. The link gravity between node j and k has been defined by regarding the node degree as the mass, the distance $\mathcal{H}_{jk}$ of the shortest path on static network G_W between j and k as the distance. The static gravity of node j can be further defined as $\sum_{k \neq j} \frac{d_j d_k}{\mathcal{H}_{jk}^2}$. The static node gravity has been used to select the seed node of an information diffusion process in order to maximize the prevalence [11], motivated by the fact that it contains both the neighborhood and the path information of a node. We generalize the gravity definition to temporal networks. The temporal link gravity of $l(j,k)$ is defined as $\frac{1}{2}(\frac{d_j d_k}{\mathcal{Q}_{jk}^2} + \frac{d_j d_k}{\mathcal{Q}_{kj}^2})$, where $\mathcal{Q}_{jk}$ is the number of links of the shortest path from j to k in all the directed spreading trees (see Figure 3.1(b)). Specifically, the shortest directed path from j to k is computed in each spreading tree rooted at one seed node. We consider the shortest among these N shortest directed paths and its length (number of links) is $\mathcal{Q}_{jk}$.

3.2.4. LINK BLOCKING AND EVALUATION

We illustrate the link blocking procedure and the evaluation method to measure the effectiveness of link blocking strategies. Given a temporal network, we specify the time window to block links as $[t_s, t_e]$. For each time window $[t_s, t_e]$, we count the number of node pairs $|\mathcal{L}_W^*(t_s, t_e)|$ that have at least one contact within $[t_s, t_e]$ and block $5\%, 10\%, 20\%, 40\%, 60\%, 80\%$ and 100% of $|\mathcal{L}_W^*(t_s, t_e)|$ links respectively using each centrality metric. The number of links to be blocked is further expressed as the fraction f of the number of links in the integrated network. For each centrality metric, we block the given fraction f of links that have the highest values for the given period $[t_s, t_e]$, i.e., remove all the contacts within $[t_s, t_e]$ associated with the selected links.

We perform the SI spreading model by setting each node as the seed node on the original temporal network as well as the temporal network after the link blocking. The average prevalence is the average over each possible seed node. The average prevalence of the SI diffusion at any time t when the selected fraction f of links are blocked within $[t_s, t_e]$ and when no links are blocked is denoted as $\rho_f(t)$ and $\rho_o(t)$ respectively, where $t \in [0, 1, ..., T]$.

The effectiveness of each centrality metric is evaluated by

$$\rho_D(f) = \frac{\sum_{t=1}^{T}(\rho_0(t) - \rho_f(t))}{\sum_{t=1}^{T}\rho_0(t)} \tag{3.1}$$

which corresponds to the area below the original prevalence $\rho_o(t)$ and above the prevalence curve $\rho_f(t)$ with link blocking normalized by the area under $\rho_o(t)$ (shown in Figure 3.2(b)). A larger $\rho_D(f)$ implies a more effective link block strategy in suppressing the SI spreading.

3.3. DATA DESCRIPTION

In this chapter, we use eight temporal network datasets to investigate the link blocking problem in temporal networks. The dataset can be classified into two categories according to the contact type, i.e., proximity (*Haggle* [13], *HighSchool2012 (HS2012)* [14], *HighSchool2013 (HS2013)* [15], *Reality Mining (RM)* [16], *Hypertext 2009 (HT2009)* [17], *Primary School (PS)* [18] and *Infectious* [17]) and electronic communication (*Manufacturing Email*(*ME*) [19]). The detailed topological features of these datasets are shown in Table 3.1, including the number of nodes, time steps, contacts, the number of links, link density, average degree and average link weight in G_W.

On each temporal network, we perform the SI spreading process starting at every node as the seed. The average prevalence ρ over time for each dataset is shown in Fig. 3.2(a), where the time step is normalized by the time span T of the observation time window. The spreading speed, i.e., how fast the prevalence grows over time, is quite different across networks. Two networks (*Haggle* and *Infectious*) show slow and relative linear increase in prevalence over times, due to the low link density in these two networks (Table 3.1). However, the prevalence in the other networks, increases dramatically at the early stage of the spreading process and converges to about 100%.

Table 3.1: Basic properties of the empirical networks. The number of nodes (N), the original length of the observation time window (T in number of steps), the total number of contacts ($|\mathscr{L}|$) and the number of links ($|\mathscr{L}_W|$), link density, average node degree ($\langle d\rangle$) and average link weight $\langle w\rangle$ in G_W are shown.

Network	N	T	$\lvert\mathscr{L}\rvert$	$\lvert\mathscr{L}_W\rvert$	link density	$\langle d\rangle$	$\langle w\rangle$
Haggle	274	15,662	28,244	2,124	0.0568	15.50	13.30
HS2012	180	11,273	45,047	2,220	0.1378	24.67	20.29
HS2013	327	7,375	188,508	5,818	0.1092	35.58	32.40
HT2009	113	5,246	20,818	2,196	0.3470	38.87	9.48
Infectious	410	1,392	17,298	2,765	0.0330	13.49	6.26
ME	167	57,791	82,876	3,250	0.2345	38.92	25.50
PS	242	3,100	125,773	8,317	0.2852	68.74	15.12
RM	96	33,452	1,086,404	2,539	0.5568	52.90	427.89

3.4. EMPIRICAL RESULTS

In this section, we evaluate the effectiveness of using aforementioned centrality metrics to select the links to be blocked within $[t_s, t_e]$. We consider diverse time windows $[t_s, t_e]$ as listed in Table 3.2. Intervention is possibly introduced at different diffusion phases. Hence,

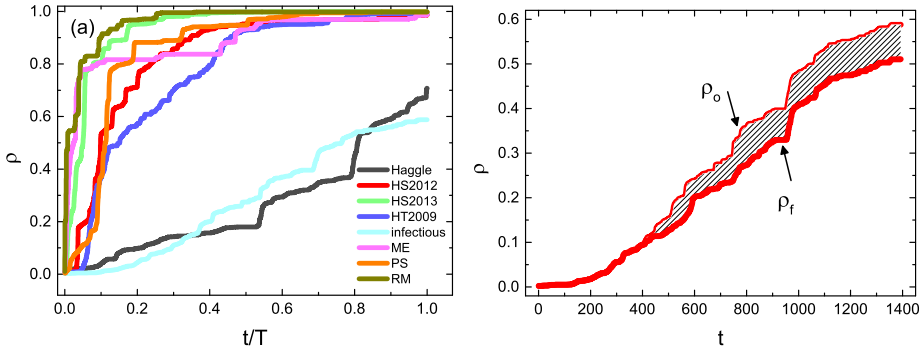


Figure 3.2: (a) Evolution of the average prevalence ρ of the SI model ($\beta = 1$) for the eight empirical datasets. (b) An example of the area difference between the original spreading curve (ρ_o) and the curve (ρ_f) after blocking f fraction of links.

$t_s \in \{T_{10\%I}, T_{20\%I}, T_{30\%I}, T_{40\%I}, T_{50\%I}\}$, where $T_{10\%I}$ is the time when the average prevalence without blocking reaches $\rho = 10\%$ (see Fig. 3.2(a)). The duration of each time window is set as the duration for the average prevalence to increase 10% just before t_s. If $t_s = T_{20\%I}$, the duration of the time window is $t_e - t_s = T_{20\%I} - T_{10\%I}$. If $t_s = T_{10\%I}$, the duration of the time window is $t_e - t_s = T_{10\%I} - T_{0\%I} = T_{10\%I}$. The number of links to block has also been chosen systematically. We take $[t_s = T_{10\%I},\ t_e = 2T_{10\%I}]$ as an example to illustrate our findings.

Table 3.2: The time window $[t_s, t_e]$ we choose for link blocking based on the average prevalence ρ when $\beta = 1$. For instance, $T_{10\%I}$ represents the time when the prevalence reaches $\rho = 0.1$.

Network	$[T_{10\%I}, 2T_{10\%I}]$	$[T_{20\%I}, 2T_{20\%I} - T_{10\%I}]$	$[T_{30\%I}, 2T_{30\%I} - T_{20\%I}]$
Haggle	[3293, 6586]	[8416, 13539]	[9523, 10630]
HS2012	[403, 806]	[675, 947]	[925, 1175]
HS2013	[50, 100]	[113, 176]	[195, 277]
HT2009	[332, 664]	[377, 422]	[439, 501]
infectious	[410, 820]	[553, 696]	[751, 949]
ME	[168, 336]	[285, 402]	[461, 637]
PS	[136, 272]	[276, 416]	[287, 298]
RM	[5, 10]	[34, 63]	[111, 188]
Network	$[T_{40\%I}, 2T_{40\%I} - T_{30\%I}]$	$[T_{50\%I}, 2T_{50\%I} - T_{40\%I}]$	
Haggle	[12440, 15357]	[12668, 12896]	
HS2012	[1043, 1161]	[1109, 1175]	
HS2013	[236, 277]	[369, 502]	
HT2009	[568, 697]	[790, 1012]	
infectious	[955, 1159]	[1062, 1169]	
ME	[731, 1001]	[1387, 2043]	
PS	[323, 359]	[347, 371]	
RM	[133, 155]	[257, 381]	

Figure 3.3 shows the effectiveness of each centrality metric as a function of f, which is the number of links blocked normalized by the number of links in the integrated network. The random selection of links from those that have at least one contact within $[t_s, t_e]$ is used as a

baseline, in which each point is the averaged over 100 realizations.

We find that four link centrality metrics always outperform the random selection: static betweenness, backbone weight, time-confined backbone weight $[t_s, t_e]$ and backbone betweenness. In *Haggle* and *infectious*, the best performance comes from static betweenness, whereas the time-confined backbone weight $[t_s, t_e]$ outperforms the other metrics in the other six networks. Figure 3.2 shows that the prevalence grows slowly over time in *Haggle* and *infectious*. Hence, the static betweenness seems a suitable link blocking strategy for networks with a slow spreading speed. However, for networks where information propagates fast, the time-confined backbone weight $[t_s, t_e]$ is a good indicator to select the links to block. Furthermore, we find that time-confined link weight $[t_s, t_e]$ outperforms link weight and time-confined backbone weight $[t_s, t_e]$ outperforms the backbone weight. This implies that considering the link temporal topological features within the blocking time window is crucial for the link selection.

For a given time window $[t_s, t_e]$, we define the average performance of a centrality metric as the area under $\rho_D(f)$ over the whole range f. The average performance is further normalized by the maximal average performance among all the centrality metrics for the given $[t_s, t_e]$. This average performance over diverse numbers of links to be blocked allows us to evaluate whether the performance of these centrality metrics is stable when the time window varies. Figure 3.4 verifies that our findings within $[t_s = T_{10\%I},\ t_e = 2T_{10\%I}]$ from Figure 3.3 can be generalized to the other time windows.

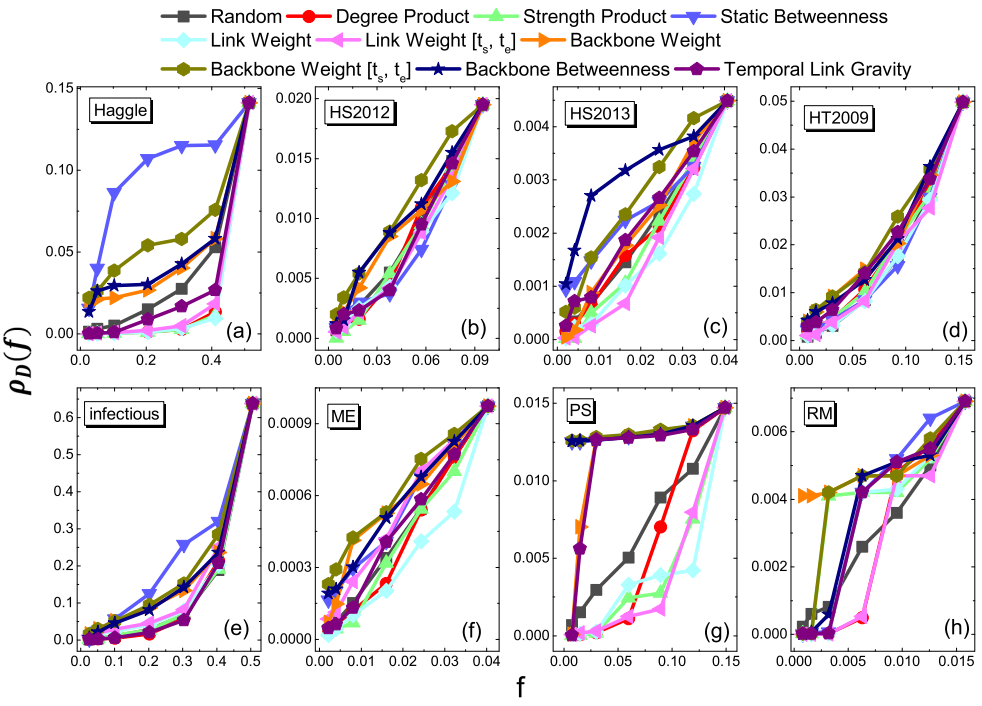


Figure 3.3: The effectiveness $\rho_D(f)$ of each centrality metric in selecting the links to block within time window $[T_{10\%I}, 2T_{10\%I}]$. Each point on the curve corresponds to block 5%, 10%, 20%, 40%, 60%, 80% and 100% of $|\mathscr{L}^*_W(t_s = T_{10\%I}, 2T_{10\%I})|$ links, respectively. The x-axis f is obtained by the number of links blocked normalized by the number of links in the integrated network.

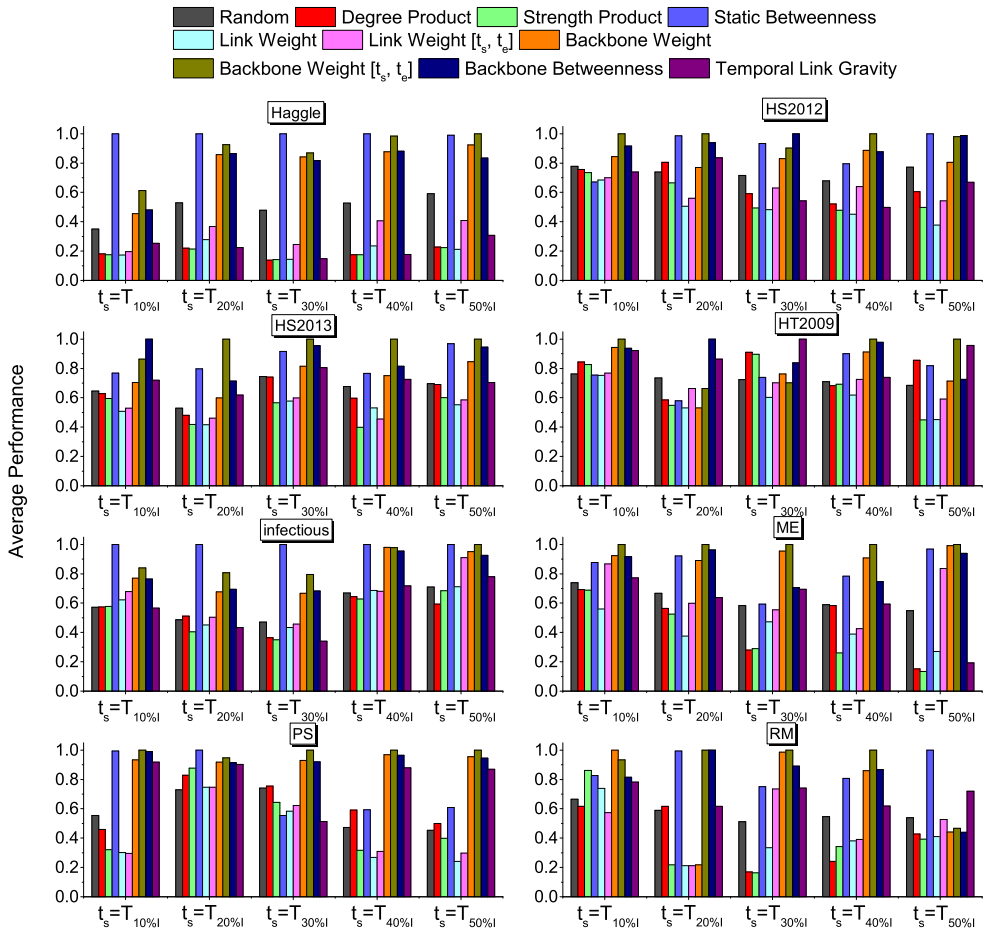


Figure 3.4: Average link blocking performance for each centrality metric over different number of blocked links, within different time windows and in different networks. The x axis shows the time windows. We only show the starting time t_s of each time window for simplicity and the ending time of each window can be found in Table 3.2.

3.5. CONCLUSION

In this chapter, we investigate how different link blocking strategies could suppress the information diffusion process on temporal networks. The spreading process is modeled by the SI model with infection probability $\beta = 1$. We propose diverse classes of link centrality metrics to capture different link temporal topological properties, including the information diffusion backbone based metrics and the static link centrality metrics. According to each metric, we select a given number of links that have the highest centrality value and block them for the given period $[t_s, t_e]$. The corresponding effect of such link blocking is evaluated via the extent that the prevalence is suppressed over time.

The empirical results from eight temporal network datasets show that four metrics outperform the random link selection, that is, backbone weight, backbone weight $[t_s, t_e]$, backbone betweenness and static betweenness. An interesting finding is that the backbone based

metrics, especially time-confined backbone weight $[t_s, t_e]$, perform well in networks where information gets prevalent fast. However, the static betweenness outperforms in networks where information propagates slowly. These observations hold for different choices of time window and the number of links to be blocked. Our findings point out the importance of both temporal and static centrality metrics in determining links' role in a diffusion process. Moreover, the time-confined metrics that explicitly explore the property/role of the contacts that occur within the time window in the global diffusion process seems promising in identifying the links to block.

In this work, we select links based on the centrality metrics that are derived from the temporal network information over the whole observation window $[0, T]$. Our study unravels actually the relation between links' or contacts' temporal topological properties and their role in a diffusion process. A more challenging question is how to identify the links to block based on the temporal network information observed so far within $[0, t_s]$.

REFERENCES

[1] X.-X. Zhan, A. Hanjalic, and H. Wang, *Information diffusion backbones in temporal networks,* Scientific reports **9**, 6798 (2019).

[2] P. Holme, *Modern temporal network theory: a colloquium,* The European Physical Journal B **88**, 234 (2015).

[3] T. Takaguchi, N. Sato, K. Yano, and N. Masuda, *Importance of individual events in temporal networks,* New Journal of Physics **14**, 093003 (2012).

[4] L. J. Peters, J.-J. Cai, and H. Wang, *Characterizing temporal bipartite networks-sequential-versus cross-tasking,* in *International Conference on Complex Networks and their Applications* (Springer, 2018) pp. 28–39.

[5] L. E. Rocha and N. Masuda, *Random walk centrality for temporal networks,* New Journal of Physics **16**, 063023 (2014).

[6] P. Grindrod, M. C. Parsons, D. J. Higham, and E. Estrada, *Communicability across evolving networks,* Physical Review E **83**, 046120 (2011).

[7] E. Estrada, *Communicability in temporal networks,* Physical Review E **88**, 042811 (2013).

[8] C. Qu, X. Zhan, G. Wang, J. Wu, and Z.-k. Zhang, *Temporal information gathering process for node ranking in time-varying networks,* Chaos: An Interdisciplinary Journal of Nonlinear Science **29**, 033116 (2019).

[9] C. Li, Q. Li, P. Van Mieghem, H. E. Stanley, and H. Wang, *Correlation between centrality metrics and their application to the opinion model,* The European Physical Journal B **88**, 65 (2015).

[10] R. Pastor-Satorras, C. Castellano, P. Van Mieghem, and A. Vespignani, *Epidemic processes in complex networks,* Reviews of modern physics **87**, 925 (2015).

[11] Z. Li, T. Ren, X. Ma, S. Liu, Y. Zhang, and T. Zhou, *Identifying influential spreaders by gravity model,* Scientific reports **9**, 8387 (2019).

[12] M. E. Newman, *Scientific collaboration networks. ii. shortest paths, weighted networks, and centrality,* Physical review E **64**, 016132 (2001).

[13] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, *Impact of human mobility on opportunistic forwarding algorithms,* IEEE Transactions on Mobile Computing , 606 (2007).

[14] J. Fournet and A. Barrat, *Contact patterns among high school students,* PloS One **9**, e107878 (2014).

[15] R. Mastrandrea, J. Fournet, and A. Barrat, *Contact patterns in a high school: a comparison between data collected using wearable sensors, contact diaries and friendship surveys,* PloS One **10**, e0136497 (2015).

[16] N. Eagle and A. S. Pentland, *Reality mining: sensing complex social systems,* Personal and ubiquitous computing **10**, 255 (2006).

[17] L. Isella, J. Stehlé, A. Barrat, C. Cattuto, J.-F. Pinton, and W. Van den Broeck, *What's in a crowd? analysis of face-to-face behavioral networks,* Journal of theoretical biology **271**, 166 (2011).

[18] J. Stehlé, N. Voirin, A. Barrat, C. Cattuto, L. Isella, J.-F. Pinton, M. Quaggiotto, W. Van den Broeck, C. Régis, B. Lina, *et al.*, *High-resolution measurements of face-to-face contact patterns in a primary school,* PloS One **6**, e23176 (2011).

[19] R. Michalski, S. Palus, and P. Kazienko, *Matching organizational structure and social network extracted from email communication,* in *International Conference on Business Information Systems* (Springer, 2011) pp. 197–206.

4

INFORMATION GATHERING PROCESS FOR INFLUENTIAL NODES IDENTIFICATION IN TEMPORAL NETWORKS

This chapter is based on the publication: C. Qu, X.-X. Zhan, G. Wang, J. Wu and Z.-K. Zhang, Temporal Information Gathering Process for Node Ranking in Time-varying Networks, Chaos: An Interdisciplinary Journal of Nonlinear Science, **29(3)**, 033116 (2019). The PhD candidate provided the initial idea, initial implementation and had significant contribution to write the paper.

In this chapter, we explore how to identify influential nodes for an information diffusion process on a temporal network. We model the information diffusion process by a susceptible-infected-recovered (SIR) model. How influential a node is is quantified by the final spreading size when the spreading process originates at the node. We propose a temporal information gathering process (TIG-process) to identify influential nodes based on the temporal network structure. The TIG-process iteratively gathers the information from the neighbors of a node as a nodal property, which is called the influence score. The neighbors of a node contains not only the direct neighbors but also nearby-nodes that are within a given distance along a path. Two types of temporal paths are introduced to identify the neighbors of a node in a temporal network, i.e., fastest arrival path and temporal shortest path. For every node, we compute the spreading size when the node is the seed node to start the spreading process which is modeled by the SIR model and also its influence score from the TIG-process. We compute the Kendall correlation coefficient between the spreading size and influence score. A higher Kendall correlation coefficient indicates a better performance of using TIG-process to identify influential nodes. We compare the identification performance of TIG-process with the benchmark metrics. Experimental results from eight temporal empirical networks show that the TIG-process using the fastest arrival path to identify neighbors outperforms the benchmark metrics. The best performance of identifying influential nodes based on the TIG-process can be achieved with a small value of n, which indicates the maximal distance we need to consider when defining a node's neighborhood, across empirical data sets. Our work sheds light on how to choose a seed for information diffusion on a temporal network to maximize the spreading size.

4.1. INTRODUCTION

A node is called influential if it can spread information widely when serving as the seed to start the spreading of a piece of information [1]. Finding influential nodes helps to promote products in viral marketing [2], to control the spread of rumors [3] and to facilitate the diffusion of useful information. Centrality metrics that describing nodal network properties, such as neighborhood-based centrality metrics and path-based centrality metrics, have been proposed to identify influential nodes in static networks [4–6]. One of the most representative neighborhood-based centrality metrics is the degree centrality. The definition of degree centrality is that nodes that have more directed neighbors are more influential. Degree centrality is a local metric with low computational complexity but has low identification power [7]. Path-based centrality metrics, such as Katz centrality [8], consider global topological structure of the network and usually show high identification performance . However, the computational complexity of these metrics is also high, and thus they are difficult to be applied to large-scale networks [1].

Despite the progress of using nodal centrality metrics to identify node's role in a dynamic process in a static network, identifying influential nodes in a temporal network is still challenging and far from well understood. [9–14]. In a temporal network, a node may play different roles at different time in a spreading process. For example, a node that only has many contacts before the information starts to spread but never has contacts during the information spreading process has no influence on the process. There are some pioneering researches concentrated on ranking influential nodes in temporal networks [15–17]. For example, some researchers first cut the temporal networks into a series of static snapshots and then estimate a node's influence using the average value of its centrality over all static snapshots [15, 16].

The nodal centrality metrics obtained in this way are the generalization of the corresponding static ones. For instance, the temporal degree, temporal closeness and temporal betweenness [15] belong to this class of methods. Even though these methods may gain some improvement in finding influential nodes compared to static metrics in temporal networks, cutting the temporal networks into static snapshots and taking the average value over all the snapshots may loss temporal information, e.g., the time order of contacts is ignored.

We aim to identify influential nodes in a temporal network. We model the information diffusion process by a discrete susceptible-infected-recovered (SIR) model. In the SIR model, an infected node transmits the information to a susceptible node with a give infection probability when they have a contact. An infected node has a probability at each time step to forget the information thus entering the recovered state, whereas the node can neither be infected or infect any other anymore. In a temporal network, a node may have contacts with the other nodes at different time steps. If we set the node as the seed node and randomly choose a time from the time steps when the node has contact(s) as the starting time of the information spreading, we can get a final spreading size for one run of the spreading process. We define the mean spreading size for a given seed node as the average over multiple final spreading sizes derived from different runs of the spreading process. The mean spreading size of a node quantifies its spreading capacity in a temporal network. A node that has a larger mean spreading size is more influential. Information diffuses along the temporal paths on a temporal network. A node with more temporal paths may better connect to the rest of the network, thus may contribute more to a spreading process. Therefore, in this chapter, we explore how temporal paths could help in identifying influential nodes. We propose a temporal information gathering process (TIG-process) to identify influential nodes for the information diffusion process. We define neighbors of a node as the nodes within a certain distance to the node along a temporal path. The TIG-process, gathers the properties not only of the given node but also of the properties of its neighbors and returns the TIG-score as a nodal property, which is further used to estimate the nodal influence in the spreading process. TIG-process is controlled by four parameters, i.e., (n, f, D^T, c), where n is the temporal gathering depth, the distance within which nodes along a temporal path are defined as neighbors, f is the weighting function, D^T is the temporal distance matrix that determines the neighbors of a node, c describes the initial TIG-score of every node. For each node, we initialize a score and then iteratively gather neighborhood information to obtain a TIG-score. For the initial score, we use classical centrality metrics, such as static degree, static closeness, node strength, static betweenness, eigenvector centrality [15, 17]. We use two kinds of temporal distance, i.e., fastest arrival distance and temporal shortest distance, to construct the distance matrix. We perform experiments of using the TIG-process to identify influential nodes on eight temporal networks. We find that the fastest arrival distance based TIG-process performs better in identifying influential nodes compared to the benchmark metrics and the TIG-process based on temporal shortest distance. TIG-process performs the best with small value of n.

The rest of the chapter is organized as follows. In Section 4.2, we first show the basic representation of a temporal network as well as notations that will be used. Then we give a detailed description of the spreading capacity of a node. Node with a higher spreading capacity is more influential for the spreading process on a temporal network. Finally, we show the TIG-process and benchmark metrics for influential node identification. The empirical datasets are given in Section 4.3 and the results are shown in Section 4.4. We discuss and conclude in Section 4.5.

4.2. Method

In this section, we first illustrate the notations that will be used in this chapter. Then, we step into how to define a node spreading capacity based on an SIR spreading model. Node with a higher spreading capacity is more influential. Finally, we propose the temporal information gathering process as well as benchmark metrics to quantify diverse nodal properties that will be used to identify influence nodes.

4.2.1. Basic notations and definitions

Let $G^T = (V, E^T)$ be a temporal network, where V is the node set, E^T is the contact set and $[1, T]$ is the observation time window. A contact $e^T \in E^T$ is given by (v_i, v_j, t), where $v_i, v_j \in V$, t represents the time when the contact happens. At each time $t \in [1, T]$, the adjacency matrix is denoted as A_t, where $A_t(i, j) = 1$ if there is a contact between node v_i and v_j at time t, otherwise, $A_t(i, j) = 0$. The unweighted aggregated static network of G^T is given by $G = (V, E)$, where E is the link set. A link exists in E if the two end nodes at least have one contact in G^T. The adjacency matrix of G is A, in which $A(i, j) = 1$ represents nodes v_i and v_j are connected, otherwise $A(i, j) = 0$. In static network, the distance between two nodes is given by length of the shortest path between them. We use the distance matrix D to denote the distance between all possible node pairs in G. An entry $D(i, j)$ of D represents the distance between the two corresponding nodes v_i and v_j.

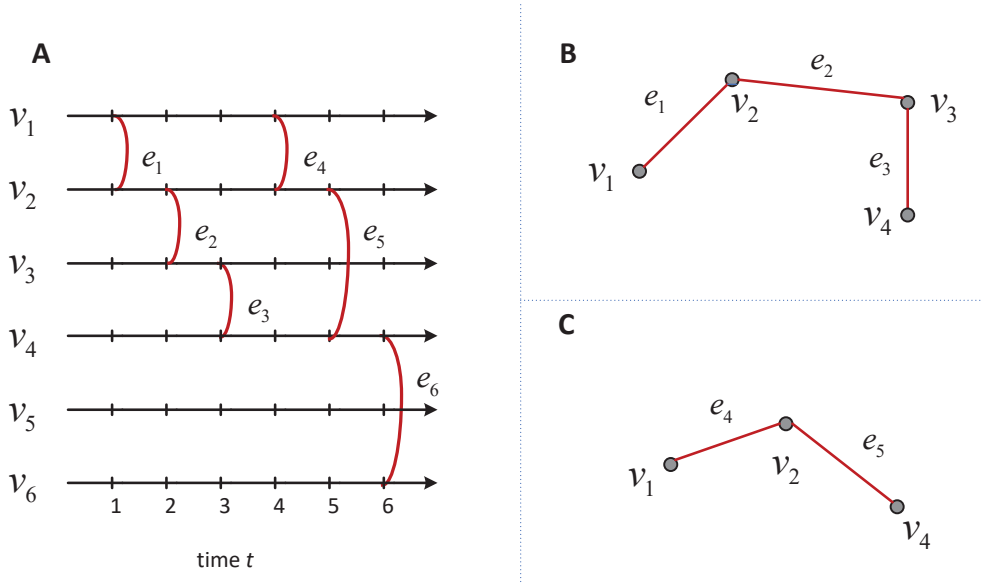


Figure 4.1: (A) A schematic representation of a temporal network with nodes $\{v_1, v_2, \ldots, v_6\}$ and contacts $\{e_1, e_2, \ldots, e_6\}$. There are two paths between node v_1 and v_4, as shown in B and C. (B) The fastest arrival path between node v_1 and v_4. (C) The shortest temporal path between node v_1 and v_4.

In a temporal network G^T, we have several ways to define the distance between nodes by considering both the topological distance along the path and the duration time of the path. We introduce two different ways of defining temporal distance between two nodes, i.e., fastest arrival distance and temporal shortest distance [18], as follows.

- **Temporal path**: A temporal path in G^T is a node sequence $P = \langle v_1, v_2, \cdots, v_k, v_{k+1} \rangle$, where $(v_i, v_{i+1}, t_i) \in E^T$ is the i-th contact on P for $1 \leq i \leq k$, and $t_i \leq t_{i+1}$, $1 \leq i < k$. The start time of P is $t_{start}(P) = t_1$ and the end time of P is $t_{end}(P) = t_k$. We define the temporal path length $l(P)$ of P as the number of links on path P. Given a time period $[t_\alpha, t_\omega]$, let $\mathbf{P}(v_i, v_j, [t_\alpha, t_\omega]) = \{P : P$ is a temporal path from v_i to v_j such that $t_{start}(P) \geq t_\alpha$ and $t_{end}(P) \leq t_\omega\}$.

- **Fastest arrival path**: The fastest arrival path between node v_i and node v_j is the path that goes from v_i to v_j taking the shortest time counted from the starting time of a temporal network, i.e., $t = 1$. In other words, the fastest arrival path is the first arrival path from the source node v_i to the target node v_j. $P \in \mathbf{P}(v_i, v_j, [1, T])$ is the fastest arrival path if $t_{end}(P) = \min\{t_{end}(P') : P' \in \mathbf{P}(v_i, v_j, [1, T])\}$. The fastest arrival distance between node v_i and node v_j is measured by the length of the fastest arrival path between them. In Figure 4.1(A), we show a temporal network with six nodes (i.e., $v_1, v_2, v_3, v_4, v_5, v_6$) and 6 contacts (i.e., $e_1, e_2, e_3, e_4, e_5, e_6$) between nodes. For nodes v_1 and v_4, there are two temporal paths between them, i.e., $P_1 = \langle v_1, v_2, v_3, v_4 \rangle$ and $P_2 = \langle v_1, v_2, v_4 \rangle$. We show P_1 and P_2 in Figure 4.1(B) and (C), respectively. The fastest arrival path between v_1 and v_4 is P_1, with $l(P_1) = 3$.

- **Temporal shortest path**: The temporal shortest path from v_i to v_j is a path for which the overall traversal time needed is the shortest. Therefore, $P \in \mathbf{P}(v_i, v_j, [t_\alpha, t_\omega])$ is a temporal shortest path if $l(P) = \min\{l(P') : P' \in \mathbf{P}(v_i, v_j, [t_\alpha, t_\omega])\}$. The temporal shortest distance between node v_i and node v_j is the length of the temporal shortest path between them. Figure 4.1(C) shows the temporal shortest path P_2 between v_1 and v_4, i.e., $l(P_2) = 2$.

- **Temporal distance matrix**: Based on the distance between nodes in a temporal network, we define the temporal distance matrix D^T, where each element $D^T(i, j)$ represents the temporal distance between node v_i and v_j. The temporal distance can be either fastest arrival distance or temporal shortest distance.

- **Distance index matrix**: For every distance value s from D^T, we define a distance index matrix D_s^T as a 0-1 matrix, where

$$D_s^T(i, j) = \begin{cases} 1 & D^T(i, j) = s \\ 0 & \text{otherwise.} \end{cases} \tag{4.1}$$

  Therefore, we have $D^T = \sum_{s=0}^{+\infty} (s \cdot D_s^T)$. Because each temporal path from a temporal network follows the time order, the distance matrix D^T and the index matrix D_s^T are both asymmetric.

4.2.2. SPREADING CAPACITY

We use discrete susceptible-infected-recovered (SIR) model to mimic information diffusion process on a temporal network. There are three states in an SIR model, i.e., susceptible (S), infected (I) and recovered (R). An infected node may infect each of its susceptible neighbors with infection probability β through the contact between them. Also, an infected node may recover to recovered state with a probability μ. Given a temporal network $G^T = (V, E^T)$, SIR

spreading process follows the time order of the contacts. For every node v_i as the seed, we illustrate how to perform the SIR spreading process to obtain its spreading capacity.

Suppose the time when node v_i has contacts with other nodes is given by set $T_{v_i} = \{t^1_{v_i}, t^2_{v_i}, \cdots, t^m_{v_i}\}$. For node v_i as the seed, we randomly select time $t^j_{v_i} \in T_{v_i} (1 \leq j \leq m)$ as the starting time of the spreading process. Thus, node v_i is initially infected at time $t^j_{v_i}$, all the other nodes are in susceptible state. We perform the SIR spreading process from $t^j_{v_i}$ until time T. The final spreading size started from node v_i at time $t^j_{v_i}$ can be denoted as $R^j_{v_i}$. At time step T, a node can be in one of the three states, i.e., susceptible, infected and recovered. Therefore, we define the final spreading size $R^j_{v_i}$ at time T as the number of infected nodes and recovered nodes.

For node v_i, we run the SIR spreading process starting from it for 1000 realizations. In each realization, we randomly choose time $t^j_{v_i}$ from T_{v_i} as the starting time of the diffusion process. Therefore, the final spreading size set from 1000 realizations can be recorded as $R(v_i) = \{R^1_{v_i}, R^2_{v_i}, \cdots, R^{1000}_{v_i}\}$. The mean spreading size $R_{mean}(i)$, given by the average over set $R(v_i)$, is defined to quantify a node's spreading capacity. A node with larger value of $R_{mean}(i)$ has larger spreading capacity.

4.2.3. TEMPORAL INFORMATION GATHERING PROCESS

In this section, we propose the temporal information gathering process, i.e., TIG-process, to measure nodal influence in the spreading process in a temporal network. Given a node v_i, we use TIG-score g_i to represent its influence score obtained from TIG-process. The process is given as follows. For each node v_i, we first assign an initial score c_i, which is viewed as the 0-order TIG-score g^0_i. Therefore, we use $g^{(0)} = (g^{(0)}_1, g^{(0)}_2, \ldots, g^{(0)}_{|V|}) = (c_1, c_2, \ldots, c_{|V|})$ to represent the initial scores for all the nodes in a network. The TIG-process is iterated based on these initial scores. The 1st-order TIG-process for each node is calculated by gathering the information from its direct neighbors, i.e., $g^{(1)} = D^T_1 g^{(0)}$. Similarly, the nth-order TIG-process for node v_i is gathering the information of its neighborhood with a distance equal to or less than n from v_i, i.e., $\mathcal{N}_{\leq n}(i)$. Thus, the nth-order TIG-process is given by

$$g^{(n)} = \sum_{s=0}^{n} f(s) \cdot D^T_s \cdot g^{(0)}, \tag{4.2}$$

where f is a function of s, which weights the influence of sth-order neighbors and D^T_s is the distance index matrix. The nth-order TIG-score is denoted by $g^{(n)}$. We use $g^{(n)}_i$ to indicate the influence score of node v_i. A larger value of $g^{(n)}_i$ implies node v_i is more influential in a spreading process in the network.

From Eq. 4.2, we know that TIG-process is determined by a quadruple (n, f, D^T, c) and the four variables are independent to each other. The variable n controls the information gathering depth, which varies from 1 to T. The weighting function f is a function of s, which weights the distance effect on the node influence. It can take different formations, such as $f_s = 1/s$ and $f_s = 1$. In this chapter, we use $f_s = 1$, which means for each node v_i, we treat all the nodes in $\mathcal{N}_{\leq n}(i)$ equally. We use the fastest arrival distance and temporal shortest distance as the temporal distance in matrix D^T, respectively. When we use fastest arrival distance, we call the TIG-process as FAD-based TIG-process (FAD-TIG for simplification). Similarly, if we use temporal shortest distance as the distance, the process is named as STD-based TIG-process (STD-TIG for simplification). The initial score c of every node is given by the benchmark metrics illustrated in next sub-section.

4.2.4. BENCHMARK METRICS

To compare with TIG-process in identifying influential nodes in a temporal network, we introduce state-of-the-art metrics that were used to measure influential nodes before. The metrics can be classified into two categories, i.e., metrics based on the corresponding aggregated static network G and metrics based on the temporal network G^T. These benchmark metrics are also used as the initial score of TIG-process.

- *Static betweenness (SB)* [19] is defined based on the aggregated static network G. The static betweenness $SB(i)$ of a node v_i is the number of shortest paths that pass through node v_i. The formula is given by

$$SB(i) = \sum_{h \neq i \neq j} \frac{\sigma_{hj}(i)}{\sigma_{hj}}, \tag{4.3}$$

  where σ_{hj} is the total number of shortest paths from v_h to v_j, and $\sigma_{hj}(i)$ is the number of shortest paths from v_h to v_j passing through v_i in a static network.

- *Static closeness (SC)* [20] of node v_i is defined on aggregated static network G and is given by the reciprocal of the sum of its distances from all the other nodes, namely

$$SC(i) = \frac{|V| - 1}{\sum_{v_j \in V \setminus v_i} D(i, j)}, \tag{4.4}$$

  where $D(i, j)$ is the distance between nodes v_i and v_j in G and $V \setminus v_i$ indicates the node set except v_i.

- *Static degree centrality (SD)* of node v_i is defined as the degree in the unweighted aggregated static network G, i.e.,

$$SD(i) = \sum_j A(i, j). \tag{4.5}$$

- *Node strength centrality (NS)* of node v_i counts the number of times node v_i has contacts with other nodes in a temporal network G_T.

$$NS(i) = \sum_{t=1}^{T} \sum_j A_t(i, j). \tag{4.6}$$

- *Static eigenvector centrality (SEC)* [21]. Given the adjacency matrix A of unweighted aggregated static network G, $SEC(v_i)$ is equal to the v_i-th component of the eigenvector corresponding to the largest eigenvalue.

4.3. DATASETS

We show the temporal network datasets that are used in this chapter, including five physical contact networks and three virtual contact networks. Some basic properties of the datasets are listed in Table 4.1. Besides, the coefficient of variation [1] of the fastest arrival distance (C_{fad}) and the temporal shortest distance (C_{std}) from each node to the others is also given in the table.

[1] The coefficient of variation is used to measure the extent of variability in relation to the mean value of a dataset, which is also known as relative standard deviation. The coefficient of variation is defined as the ratio of the standard deviation to the mean: $C = \frac{standard\ deviation}{mean\ value}$.

Table 4.1: Basic properties of the temporal networks. We show the number of nodes ($|V|$), the length of the observation time window (T), the total number of contacts ($|E^T|$). In addition, C_{fad} denotes the coefficient of variation of the fastest arrival distance from each node to the others. C_{std} indicates the coefficient of variation of the temporal shortest distance from each node to the others.

Network	$\|V\|$	T	$\|E^T\|$	C_{fad}	C_{std}
HS2011	126	42	28,561	0.5798	0.3405
HS2012	180	87	45,047	0.6196	0.3664
PS	242	20	125,773	0.5288	0.1188
WP	92	108	9,827	0.6191	0.4102
HC	75	90	32,424	0.8411	0.7956
EEC	771	68	38,328	1.2913	0.6522
ME	167	268	82,927	0.9081	0.6629
Collegemsg	1898	188	61,726	2.4579	1.0645

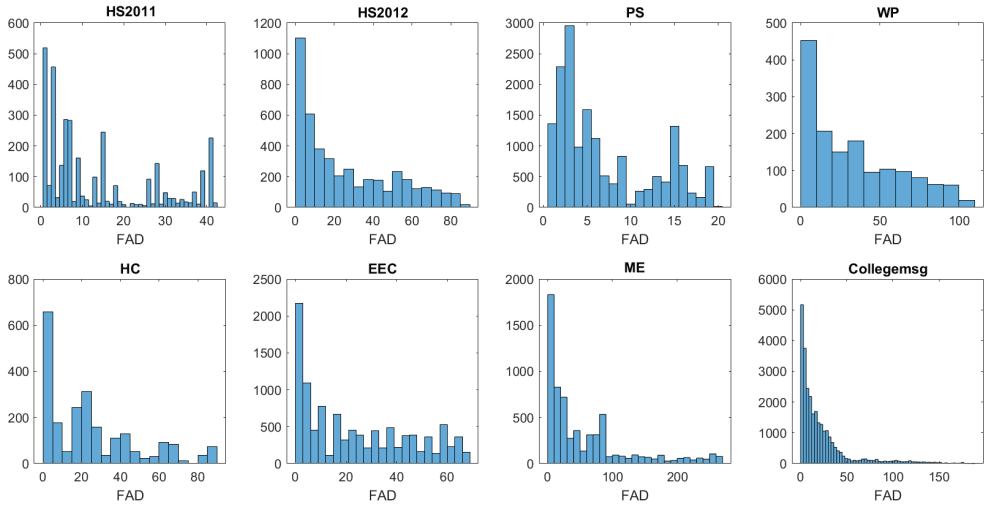


Figure 4.2: Distribution of the fastest arrival distance (FAD) between nodes in each dataset.

- High School 2011 (2012) (*HS2011*, *HS2012*) [22]. The datasets record the contacts between individuals in a high school in Marseilles, France.
- Primary School (*PS*) [23]. The dataset records the contacts between individuals in a primary school.
- Workplace (*WP*) [24]. The dataset contains contacts between individuals in an office building.
- Hospital ward contact (*HC*) [25]. The dataset records contacts between individuals in a hospital ward in Lyon, France.
- Email-Eu-Core (*EEC*) [26]. The dataset records email contacts between individuals from a large European research institution.

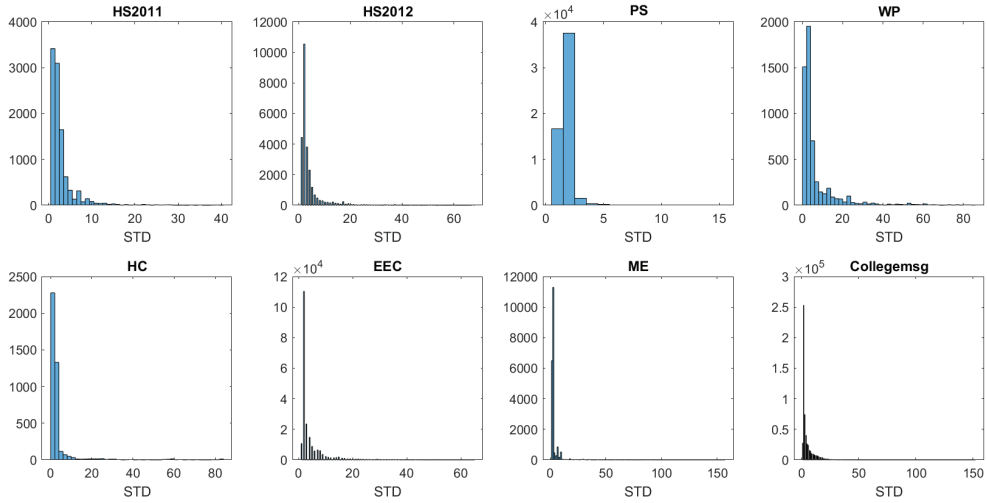


Figure 4.3: Distribution of the temporal shortest distance (STD) between nodes in each datasets.

- Manufacturing emails (*ME*) [27]. The dataset records email contacts between employees of a mid-size manufacturing company.
- CollegeMsg (*Collegemsg*) [28]. This network records contacts between individuals at the University of California, Irvine.

The fastest arrival distance (FAD) and temporal shortest distance (STD) are the two types of distance we use in the TIG-process. We give the distribution of the fastest arrival distance (FAD) and temporal shortest distance (STD) in Figure 4.2 and Figure 4.3, respectively. For a given network, the distribution of STD is heterogeneous, indicating that most of the nodes tend to have short temporal shortest distance. However, the distribution of the fastest arrival distance is more homogeneous. This implies if we use the fastest arrival paths to define a node's neighbor, the node tends to have high-order neighbors.

4.4. RESULTS

We evaluate the performance of TIG-process as well as the benchmark metrics in identifying influential nodes for a spreading process. The real spreading capacity of a node v_i is given by the mean spreading size $R_{mean}(i)$ conducted by the SIR spreading process. Taking TIG-process as an example, we illustrate how to evaluate its performance. We first get a TIG-score list by computing each node's TIG-score. Then we conduct the SIR model on the network to obtain each node's mean spreading size. Therefore, we compute the Kendall ranking correlation coefficient [2] τ between the TIG-score list and mean spreading size list. The higher value of τ indicates the better a node ranking metric in identifying influential nodes.

We use static betweenness (SB), static closeness (SC), static degree (SD), Node strength (NS) and static eigenvector centrality (SEC), respectively, as the initial score of the TIG-process. Recall that FAD-TIG and STD-TIG are two kinds of TIG-process based on the temporal distance, i.e., fastest arrival distance and temporal shortest distance, respectively. We set the infection probability $\beta = 0.1$ and recovery probability $\mu = 0.01$ for the SIR model to guarantee the spreading process can spread out from the seed node.

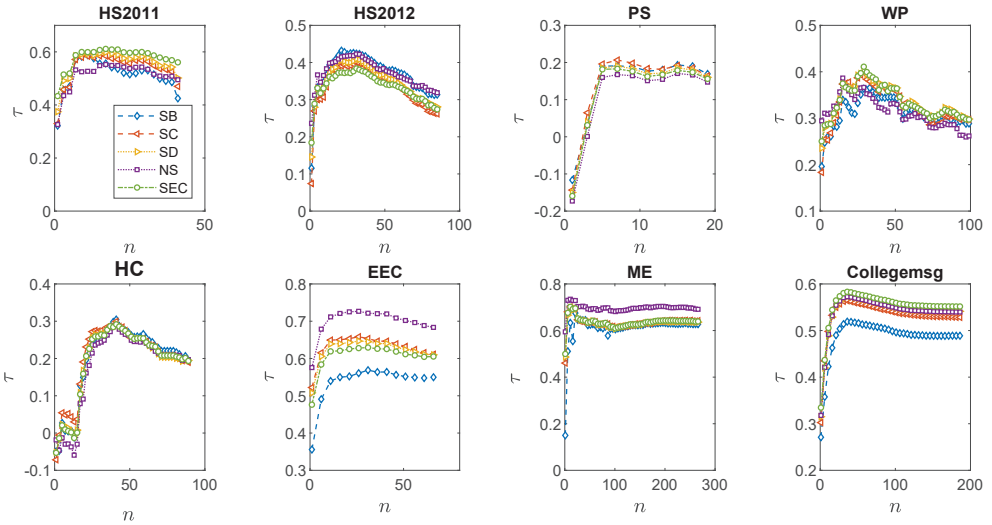


Figure 4.4: The evolution of Kendall correlation coefficient between FAD-TIG-score and R_{mean} with the increasing of information gathering depth n for eight temporal network datasets. Different colors indicate using different benchmark metrics as the initial score for the TIG-process. When $n = 0$, the Kendall correlation is between the influential score derived from the benchmark metrics and the mean spreading size.

The results are shown in Figure 4.4 and Figure 4.5 for FAD-TIG and STD-TIG, respectively. The Kendall ranking correlation coefficient between FAD-TIG score (STD-TIG score) and R_{mean} is denoted as τ. We show how the gathering depth n and the initial score of the

[2]The Kendall correlation coefficient τ is defined as follows. Let $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ be the observations of two joint random variables X and Y. Then Kendall rank correlation coefficient $\tau \in [-1, 1]$ is defined as

$$\tau = \frac{1}{n(n-1)} \sum_{i \neq j} \text{sgn}(x_i - x_j)\text{sgn}(y_i - y_j). \tag{4.7}$$

If τ takes the value of $+1$, then the agreement of the two rankings is perfect. If τ is -1, then one list is the reverse of the other. If τ is close to zero, then the two rankings are independent.

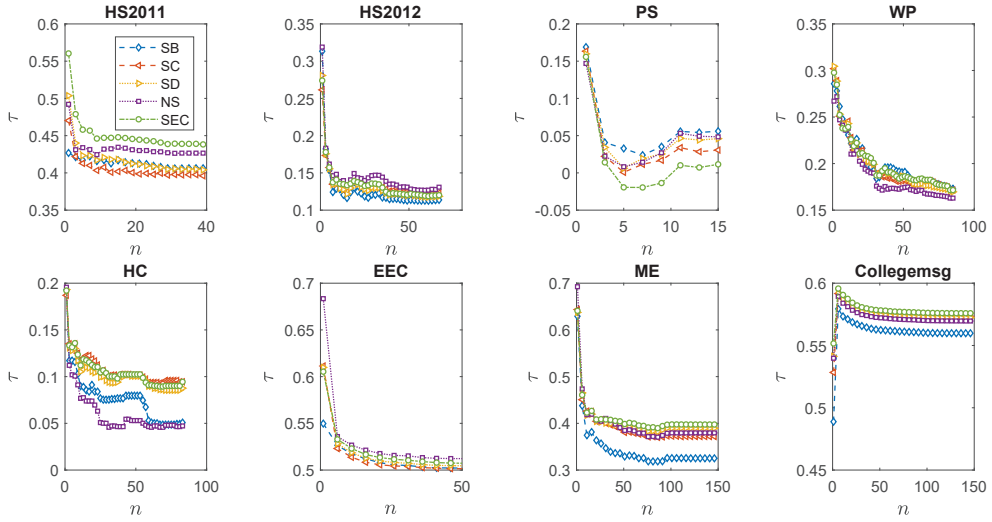


Figure 4.5: The evolution of Kendall correlation coefficient between STD-TIG-score and R_{mean} with the increasing of gathering depth n for eight temporal network datasets. Different colors indicate using different benchmark metrics as the initial score for the TIG-process. When $n = 0$, the Kendall correlation is between the influential score derived from the benchmark metrics and the mean spreading size..

TIG-process affect the performance of identifying influential nodes. When we use the fastest arrival distance as the temporal distance matrix (Figure 4.4), the Kendall ranking correlation between the FAD-TIG-score and R_{mean} increases with the information gathering depth n when n is small. Actually, the information gathering depth $n = 0$ corresponds to use each of the benchmark metrics directly as the influence score. The increase of τ with n indicates that FAD-TIG performs better than the benchmark metrics in identifying influential nodes. Also, the information gathering from the high-order neighbors generated by the fastest arrival path can help to improve the identification performance. We find that, in different networks, the best performance varies with the use of initial score. For example, TIG-process with *SEC* performs the best in networks *HS2011*, *WP* and *Collegemsg*. But TIG-process with *NS* performs the best in networks *EEC* and *ME*. The highest Kendall correlation value τ can be achieved with small n almost in all the networks with different settings of initial score. In Figure 4.5, we show the results of using temporal shortest distance as the temporal distance matrix. In seven networks, τ decreases with the increase of n, which implies the STD-TIG performs worse than the benchmark metrics. In *Collegemsg*, STD-TIG performs better than the benchmark metrics. Taking Figure 4.4 and Figure 4.5 together, FAD-TIG shows better performance than STD-TIG as well as the benchmark metrics. Thus, using the fastest arrival distance as the temporal distance matrix in the TIG-process can better identify influential nodes for the spreading process in a temporal network.

The assumption of the information gathering process is based on the fact that the influence of the nodes is related to their neighbors in a temporal network, not only immediate neighbors but also higher-order neighbors. Therefore, when n is small, we are gathering information from neighboring nodes that are close to the current node both in time and in the number of hop count. When n is large, neighboring nodes that are far away are also included. Therefore, the decrease of the performance when n is large implies that the neighbors that

are far away from the current node have a small influence on its influence ranking. Based on the distribution of FAD and STD given in Section 4.3, we find that if we use the fastest arrival distance, a node tends to have more high-order neighbors compared to that of using temporal shortest distance. Therefore, the TIG-process based on the fastest arrival distance can gather more high-order neighbor information for a node. This may explain the better performance of TIG-process based on the fastest arrival distance.

4.5. CONCLUSION

The evolution of temporal network over time makes the static nodal properties fail to identify influential nodes in a temporal network.

We proposed a temporal information gathering (TIG) process to identify influential nodes in temporal networks. The assumption is that the spreading capacity of a node relies on its neighbors. We observe that the fastest arrival distance based TIG process performs much better than the one based on temporal shortest distance as well as all the bench mark nodal centrality metrics. In addition, there is an optimal gathering depth n which makes FAD based TIG-process perform the best. Our work sheds light on how to characterize properties of a node by considering the time order of the contacts. We consider using the temporal distance between nodes as the distance matrix in TIG process. In the future work, it's interesting to use other matrix,such as Laplacian matrix, to explore whether we can gain more improvement in identifying nodes that are influential in the spreading process.

REFERENCES

[1] L. Lü, D. Chen, X.-L. Ren, Q.-M. Zhang, Y.-C. Zhang, and T. Zhou, *Vital nodes identification in complex networks,* Physics Reports **650**, 1 (2016).

[2] Z. Zhu, *Discovering the influential users oriented to viral marketing based on online social networks,* Physica A: Statistical Mechanics and its Applications **392**, 3459 (2013).

[3] N. Zhou, X.-X. Zhan, Q. Ma, S. Lin, J. Zhang, and Z.-K. Zhang, *Identifying spreading sources and influential nodes of hot events on social networks,* in *International Workshop on Complex Networks and their Applications* (Springer, 2017) pp. 946–954.

[4] S. Xu, P. Wang, and J. Lü, *Iterative neighbour-information gathering for ranking nodes in complex networks,* Scientific reports **7**, 41321 (2017).

[5] J. G. Restrepo, E. Ott, and B. R. Hunt, *Characterizing the dynamical importance of network nodes and links,* Physical review letters **97**, 094102 (2006).

[6] C. M. Taniguchi, B. Emanuelli, and C. R. Kahn, *Critical nodes in signalling pathways: insights into insulin action,* Nature reviews Molecular cell biology **7**, 85 (2006).

[7] A. Landherr, B. Friedl, and J. Heidemann, *A critical review of centrality measures in social networks,* Business & Information Systems Engineering **2**, 371 (2010).

[8] L. Katz, *A new status index derived from sociometric analysis,* Psychometrika **18**, 39 (1953).

[9] P. Holme and J. Saramäki, *Temporal networks,* Physics reports **519**, 97 (2012).

[10] F. Kuhn and R. Oshman, *Dynamic networks: models and algorithms,* ACM SIGACT News **42**, 82 (2011).

[11] D. G. Rand, S. Arbesman, and N. A. Christakis, *Dynamic social networks promote cooperation in experiments with humans,* Proceedings of the National Academy of Sciences **108**, 19193 (2011).

[12] M. G. Zimmermann, V. M. Eguíluz, and M. San Miguel, *Coevolution of dynamical states and interactions in dynamic networks,* Physical Review E **69**, 065102 (2004).

[13] S. Aral, L. Muchnik, and A. Sundararajan, *Distinguishing influence-based contagion from homophily-driven diffusion in dynamic networks,* Proceedings of the National Academy of Sciences **106**, 21544 (2009).

[14] X.-X. Zhan, A. Hanjalic, and H. Wang, *Information diffusion backbones in temporal networks,* arXiv preprint arXiv:1804.09483 (2018).

[15] H. Kim and R. Anderson, *Temporal node centrality in complex networks,* Physical Review E **85**, 026107 (2012).

[16] R. K. Pan and J. Saramäki, *Path lengths, correlations, and centrality in temporal networks,* Physical Review E **84**, 016105 (2011).

[17] T. Takaguchi, N. Sato, K. Yano, and N. Masuda, *Importance of individual events in temporal networks,* New Journal of Physics **14**, 093003 (2012).

[18] H. Wu, J. Cheng, S. Huang, Y. Ke, Y. Lu, and Y. Xu, *Path problems in temporal graphs,* Proceedings of the VLDB Endowment **7**, 721 (2014).

[19] M. E. Newman, *Scientific collaboration networks. ii. shortest paths, weighted networks, and centrality,* Physical review E **64**, 016132 (2001).

[20] G. Sabidussi, *The centrality index of a graph,* Psychometrika **31**, 581 (1966).

[21] M. E. Newman, *Mathematics of networks,* The new Palgrave dictionary of economics , 1 (2016).

[22] J. Fournet and A. Barrat, *Contact patterns among high school students,* PloS One **9**, e107878 (2014).

[23] J. Stehlé, N. Voirin, A. Barrat, C. Cattuto, L. Isella, J.-F. Pinton, M. Quaggiotto, W. Van den Broeck, C. Régis, B. Lina, *et al.*, *High-resolution measurements of face-to-face contact patterns in a primary school,* PloS One **6**, e23176 (2011).

[24] M. Génois, C. L. Vestergaard, J. Fournet, A. Panisson, I. Bonmarin, and A. Barrat, *Data on face-to-face contacts in an office building suggest a low-cost vaccination strategy based on community linkers,* Network Science **3**, 326 (2015).

[25] P. Vanhems, A. Barrat, C. Cattuto, J.-F. Pinton, N. Khanafer, C. Régis, B.-a. Kim, B. Comte, and N. Voirin, *Estimating potential infection transmission routes in hospital wards using wearable proximity sensors,* PloS one **8**, e73970 (2013).

[26] A. Paranjape, A. R. Benson, and J. Leskovec, *Motifs in temporal networks,* in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining* (ACM, 2017) pp. 601–610.

[27] R. Michalski, S. Palus, and P. Kazienko, *Matching organizational structure and social network extracted from email communication,* in *International Conference on Business Information Systems* (Springer, 2011) pp. 197–206.

[28] P. Panzarasa, T. Opsahl, and K. M. Carley, *Patterns and dynamics of users' behavior and interaction: Network analysis of an online community,* Journal of the American Society for Information Science and Technology **60**, 911 (2009).

5

DEGREE-BIASED RANDOM WALK FOR LARGE-SCALE NETWORK EMBEDDING

This chapter have been published as: Y. Zhang, Z Shi, D Feng and X.-X. Zhan*, Degree-biased random walk for large-scale network embedding, Future Generation Computer Systems, **100**, 198-209 (2019). The PhD candidate contributed to the plan and supervision of the experiments, and writing the paper.

Network embedding aims at learning node representation by preserving the network topology. Previous random-walk-based network embedding algorithms utilize random walk to sample walks and generate node pairs from the walks. The node pair set is further used as the input for a Skip-Gram model, which embeds nodes into vectors. However, these algorithms do not scale for large real-world networks which usually contain millions of nodes. They generally sample equal number of walks from every source node and set the length of every walk to be the same. This kind of setting results in a large amount of redundant node pairs as the input for the Skip-Gram model. In this chapter, we propose DiaRW, a scalable network embedding method based on a degree-biased random walk. In the degree-biased random walk, we allow a walker to visit more around the local structure of high degree nodes. Also, walks that start from high degree source nodes have longer length. DiaRW greatly reduces the size of node pair set, which is efficient for large-scale network embedding. Empirical experiments on node classification and link prediction show that DiaRW outperforms baseline embedding algorithms on a variety of real-world networks. Furthermore, our algorithm is able to learn the network embedding on networks with millions of nodes and edges in hours on a single machine, which is tenfold faster than previous algorithms.

5

5.1. INTRODUCTION

Networks have been widely used to represent components (nodes) and their interactions (links) in various areas including social science (social networks) [1], linguistics (semantic web) [2], Internet of Things (sensor network) [3] and biology (Protein-Protein interaction network) [4]. The scale of complex networks ranges from hundreds to billions of nodes, leading to a problem of how to analyze large networks in an efficient way. Network embedding, which maps each node to a low-dimensional vector, provides a ubiquitous way to represent and thus analyze networks [5, 6]. Given a network, it is often desirable to extract latent information associated with each node by learning algorithms. The latent information contains a variety of properties of the original network. For example, it may preserve the local neighborhood structure of every node as well as global community structure of the network. The embeddings are further used as features for network analysis and tasks such as node classification [7], clustering [8, 9], link prediction [10, 11], and visualization [12, 13].

Despite the enormous potential of network embedding, we argue that there exists two main challenges:

High non-linearity: As stated in [14], most network data is often sophisticated and the underlying structure of it is highly non-linear. Therefore, the embedding algorithms, such as principle component analysis (PCA) and multidimensional scaling (MDS), that embed network into a linear space via linear transformations, cannot well preserve the network structure.

Scalability: With the arrival of the age of big data, the scale of real-world networks is exploding. Taking social network as an example, the Twitter network contains 175 million active users and approximately twenty billion edges in 2012 [15]. Therefore, when it comes to large-scale networks, the massive learning task may cost months, or even simply fail due to insufficient memory, which is practically unfeasible for practical applications.

Among the network embedding algorithms that embed network data in a non-linear manner, random-walk-based network embedding algorithms have been recently proposed and show high efficiency and effectiveness. The general random-walk-based network embedding algorithms usually take two steps: In the first step, a sampling method, such as random

walk, which explores node proximity is performed to get the walk sequences. We further extract node pairs from the walks. Then, a language learning model, such as Skip-Gram [16], is applied to obtain the embedding vector for each node by using the node pair set as input. *DeepWalk* [5] is the pioneer work in using random walks to learn node representations. *Node2Vec* [6] further introduces a biased random walk procedure which combines BFS and DFS style neighborhood exploration.

Real-world networks are generally scale-free, i.e., most nodes have a low degree while only a few have high degrees. The heterogeneity of node degree implies that the local structure surrounding nodes with different degrees can vary a lot. However, the random walk strategies proposed previously usually sample the same number of walks from every source node. On the one hand, given a fix number of walks starting from every source node, walks starting from source nodes that have much larger degree than the number of walks may be not possible to visit most of the surrounding structure of the source nodes. On the other hand, walks starts from source nodes have much lower degree than the number of walks may oversample the surrounding structure of the source nodes. Additionally, previous random walk strategies also set the walks to have the same length. This setting may generate many redundant node pairs as input for the subsequent learning model. For example, for a pair of nodes that are only connected to each other, both of the two end nodes have degree 1. If we set the length of the walk starting from one of the two end nodes as 80 (the same as *DeepWalk* and *Node2Vec*), we will get a walk only contains these two end nodes, in which each node is repeated for 40 times.

To further demonstrate the inference above, we use Barabási-Albert (BA) model [17] to generate a scale-free network with 2^{16} nodes and plot the degree distribution of nodes in Figure 5.1(a). We find that the original network follows a standard power-law distribution with a slope -2.67. By contrast, we also plot the degree distribution of nodes in a node pair set generated by the uniform random walk used in *DeepWalk* in Figure 5.1(b). We observe that the degree distribution generated by the uniform random walk differs significantly from the real degree distribution. The results indicate that the node pairs sampled by using uniform random walk cannot well preserve the network properties, such as degree distribution. This may result in the embeddings obtained from *DeepWalk* cannot well preserve the network structure.

To better preserve network topology without generating a lot of redundant node pairs, we propose a high-degree biased variable-length random walk embedding (*DiaRW*) algorithm, which considers the heterogeneity of node degree. To be specific, we allow a walk to step back to nodes with a higher degree more likely in a probabilistic way, which means high-degree nodes tend to be revisited more. This also allows walks starting from a high-degree nodes to travel the local surrounding structure of the node more. Moreover, instead of setting a fixed length for all the walks starting from different source nodes, we set the length of walks starting from every source node based on the source node's centrality to avoid generating redundant node pairs. Taking degree centrality as an example, we set the length of walks that start from a high-degree source node to be longer. Experimental results on node classification and link prediction indicate that *DiaRW* show better performance compared to the baseline models. Additionally, the reduction of redundant node pairs by the degree-biased random walk makes *DiaRW* scalable for large-scale networks.

The rest of the chapter is organized as follows. In Section 5.2, we first give a review of the related work. Then, we propose the *DiaRW* algorithm in Section 5.3. In Section 5.4, we empirically evaluate our method on prediction tasks, i.e., node classification and link

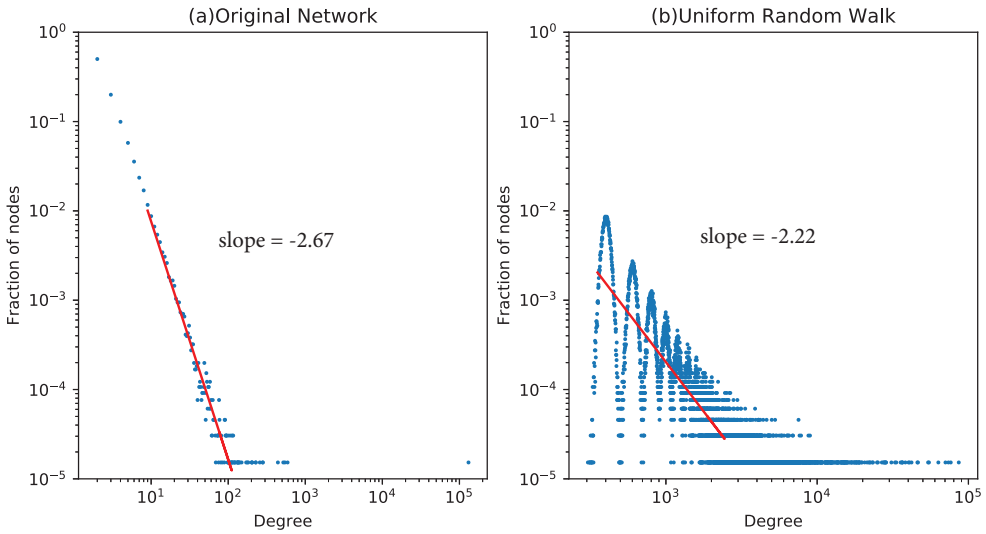


Figure 5.1: The degree distributions of (a) the original BA network with 2^{16} nodes and (b) the node pair set generated by uniform random walk from the original network.

prediction, on large-scale networks and analyze the parameter sensitivity as well as scalability of our algorithm. The chapter is concluded in the Section 5.5.

5.2. RELATED WORK

Network representation has become an important way to represent and analyze complex network. The algorithms aim to learn embedding vectors for nodes in the network. The embeddings are further used for network tasks such as link prediction, classification, etc. The learning algorithms can be categorized into two types: matrix factorization (MF)-based and neural network-based [18].

MF-based methods are either linear [19] or nonlinear [20] in learning node embeddings. The former employs the linear transformations to embed nodes into a low dimensional embedding space, such as singular value decomposition (SVD) and multiple dimensional scaling (MDS) [19]. The latter maps nodes into a low dimensional latent space by utilizing the nonlinear transformations, e.g., kernel PCA [21], spectral embedding, marginal fisher analysis (MFA) [22], and manifold-learning approaches including LLE [23] and ISOMAP [20]. Generally speaking, MF-based methods have two main drawbacks: (1) Due to the eigen-decomposition operations on data matrices, they are usually computationally expensive and are difficult to be applied on large-scale network data [24, 25]; (2) the performance is rather sensitive to the predefined proximity measures for calculating the affinity matrix.

Neural network-based methods are the state-of-the-art node representation learning techniques. The pioneer work *DeepWalk* [5] extended the idea of Word2Vec [16] to embed a network. *Node2Vec* algorithm [6] is considered as an extension of *DeepWalk*, introducing a biased random walk sampling strategy which combines BFS-style and DFS-style neighborhood exploration. However, both of them adopted a global walk strategy which ignores individual heterogeneity. Another shortcoming of *Node2Vec* is that its second-order random walks take too much time to compute the interconnections between neighbors of every node.

There are some follow-up works exploiting both 1st-order and 2nd-order proximity between nodes to embed networks. Specifically, *LINE* [26] derives a joint optimization function for preserving the first and the second-order proximity. It performs the optimization by stochastic gradient descent with edge sampling, aiming at efficiently embedding large-scale networks. The goal is similar as our chapter, nevertheless, the performance tends to be inferior compared to ours due to its limitation and inflexibility for low order proximity. *HOPE* [27] defines some similarity measures between nodes which are helpful for preserving higher-order proximity as well and formulates those measures as a product of sparse matrices to efficiently find the latent representations. However, the algorithm showed poor scalability for large-scale networks.

Table 5.1: Table of notations

$G = (V, E)$	A network G with set V of nodes and set E of edges
$\Phi(u)$	embedding vector of node u
$N_S(u)$	Neighbor set of node u generated by a sampling strategy S
$p(u, v)$	Transit probability from node u to node v
$r(v, u)$	Probability of backtracking from node v to node u
$S(u, v)$	Similarity score of node u and node v
$d(u)$	Degree of node u
$L_{\max}$	Upper bound of walk length, the default is 80.
$L(u)$	the length of a walk starting at source node u
k	the number of random walks starting from a node, the default is 10.
d	Dimension of embedding vector, the default is 128.
w	Window size while generating node pairs from a walk, the default is 10.

5.3. PROPOSED METHOD

In this section, we introduce *DiaRW*, a network embedding algorithm based on Skip-Gram model. In fact, the efficiency of the Skip-Gram based algorithms largely depends on the sampling strategies. In our algorithm, we propose a high-degree biased backtracking random walk to sample walks from a network. To reduce redundancy in the sampling process, the length of each walk is determined by the centrality value of the source node.

A network is defined as $G = (V, E)$, where V is node set and E is the set of edges. The number of nodes in a network is denoted as $N = |V|$. Table 5.1 includes notations that will be used throughout this chapter.

5.3.1. NETWORK EMBEDDING FRAMEWORK

Network embedding aims to learn a mapping function Φ: $V \rightarrow R^d (d \ll |V|)$, i.e., to embed each node in a network to a d-dimensional space. We use $\Phi(u)$ to represent the embedding vector of node u, and d is the dimension of $\Phi(u)$. The function Φ preserves network topology, such that two nodes which are close in the original network should also be close in the embedding space.

Inspired by [5, 6], we formulate network embedding as a maximum likelihood optimization problem. For every center node $u \in V$, we define the neighbor set of u as $N_S(u)$, which is generated by a sampling strategy S. For example, $N_S(u)$ is a set of nodes within a distance w from u in one walk sequence. Therefore, we give the objective function that need to be optimized as:

$$\operatorname{argmax} \sum_{u \in V} \log \Pr(N_S(u) | \Phi(u)) \tag{5.1}$$

Skip-Gram [16] is a learning model that maximizes the co-occurrence probability among the nodes that appear within a window size w, in a walk sequence. It approximates the conditional probability in Eq. (5.1) by assuming that the likelihood of observing a neighborhood node is independent of observing any other neighborhood node given the embedding vector of the source node, which is expressed as follows:

$$\Pr(N_S(u) | f(u)) = \prod_{n_i \in N_S(u)} \Pr(n_i | \Phi(u)) \tag{5.2}$$

The conditional likelihood $Pr(n_i|\Phi(u))$ is further modeled by softmax function as follows:

$$\Pr(n_i | \Phi(u)) = \frac{\exp(\Phi(n_i)\Phi(u))}{\sum_{v \in V} \exp(\Phi(v)\Phi(u))}, \tag{5.3}$$

As computing $\sum_{v \in V} \exp(\Phi(v)\Phi(u))$ is very expensive, we use the negative sampling method [16] to speed up training. We show how to learn node embeddings by Skip-Gram model in Algorithm 1 [5]. We map each node u from walk sequences to its current embedding vector $\Phi(u)$. To maximize the probability of observing its neighbors given the embedding vector of u, we use stochastic gradient descent to iteratively update it (line 3-4).

Algorithm 1: Skip-Gram (Φ, walks, w)

Input: matrix of node representations Φ
walk sequences walks
window size w
1 **for** each node $u \in$ walks and its index as inx_u **do**
2 **for** each node $n_i \in$ walks $[\text{inx}_u - w, \text{inx}_u + w]$ **do**
3 $J(\Phi) = -\log \Pr(n_i | \Phi(u))$
4 $\Phi = \Phi - \alpha \frac{\partial J}{\partial \Phi}$
5 **end for**
6 **end for**

5.3.2. SCALE-FREE NETWORKS

A scale-free network is a network whose degree distribution follows a power law, or at least asymptotically. Most of real-world networks are reported to be scale-free [17], from web graphs to social networks, protein networks and semantic networks.

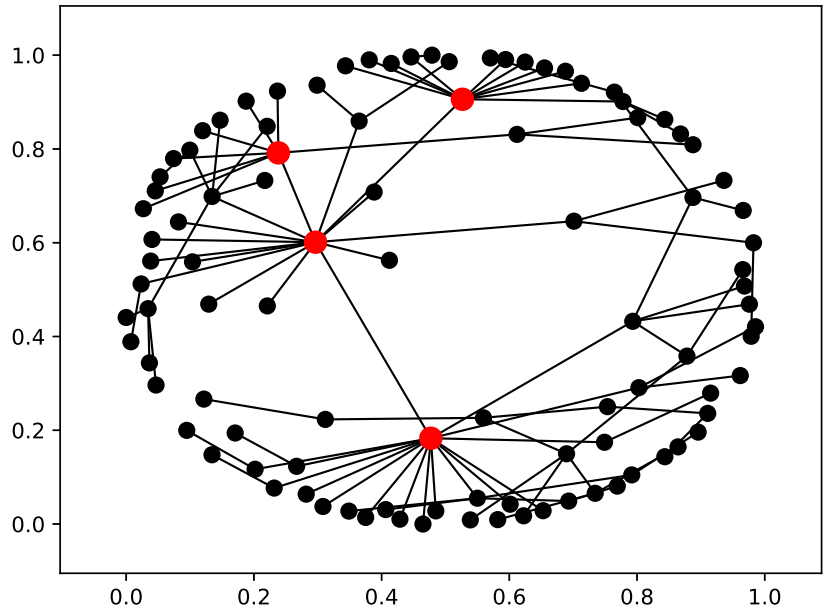


Figure 5.2: A small example of scale-free network (nodes colored red are hubs).

In Figure 5.2, we give a toy example of scale-free network, where most nodes have a low degree but some have a very high degree. Nodes with a number of edges that greatly exceeds the average are called hubs (red color nodes in Figure 5.2). Hubs usually play important

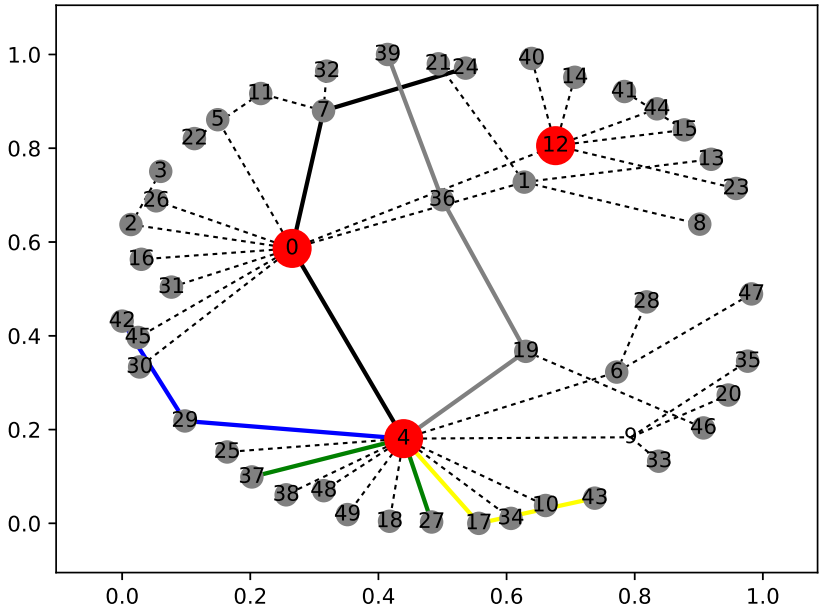


Figure 5.3: An example of uniform random walk sampling on a scale-free network. It starts from hub node 4 with the number of walks 5 and walk length 4, generating node sequences(shown in solid line) like “4-29-42-29”(blue), “4-0-7-24”(black), “4-27-4-37”(green), “4-19-36-39”(grey) and “4-17-43-17”(yellow).

roles in the network. For example, hub nodes promote information diffusion [28, 29], and we can control the epidemic spreading by isolating infected hub nodes from the susceptible population [30]. In Figure 5.3, we give a toy example of how uniform random walk fails to extract the local structure surrounding hub nodes. We set the number of walk to be 5 and the walk length to be 4. Taking the walks starting from hub node 4 as an example, we get walks such as "4-29-42-29"(blue), "4-0-7-24"(black), "4-27-4-37"(green), "4-19-36-39"(grey) and "4-17-43-17"(yellow). Within this setting, we find that the walks only visit a few immediate neighbors of node 4. The setting may make the learning model fails to learn the proximity of node 4 with most of its immediate neighbors. This problem could be more serious for real-world networks. Because real-world networks contain nodes with degrees of tens of thousands whose structure is hard to be well extracted under acceptable sample size. The local structure surrounding nodes with high and low degree may differ dramatically, thus we need to treat them differently while performing the sampling strategy.

5.3.3. SAMPLING STRATEGY

In this section, we will introduce a high-degree biased variable-length random walk sampling strategy to generate walks from a network.

HIGH-DEGREE BIASED BACKTRACKING

According to the previous section, uniform random walk cannot well adapt to the scale-free characteristic of real world networks due to the under-sampling of hubs. To solve this problem, we propose a high-degree node biased sampling random walk. We define our high-degree biased backtracking mechanism in the following way. Given a node u and its direct neighbor node v in a network, we define $p(u, v)$ as the transition probability from u to v, and $r(v, u)$ as the return probability from v to u in the random walk. The probabilities $p(u, v)$ and $r(v, u)$ are given by

$$p(u, v) = \frac{1}{d(u)}, \tag{5.4}$$

$$r(v, u) = \max\left\{0, 1 - \frac{d(v)}{d(u)}\right\} \tag{5.5}$$

where $d(u)$ represents degree of node u. Suppose a walker now is sitting at node u, the definition of $p(u, v)$ allows the walker to visit each of node u's neighbors with equal probability $p(u, v)$. If a walker is sitting at node v currently, the definition of $r(v, u)$ allows a walker to tend to revisit upstream node u if $d(u) > d(v)$. To be specific, the probability of backtracking to node u is $r(v, u)$. The walker has probability of $1 - r(v, u)$ to stay at node v. From a macro perspective, this strategy simulates a BFS-like explorations by restricting search from high-degree nodes to local structure and a DFS-like explorations by moving further away from low-degree nodes, thus better capturing the local and global network structure.

VARIABLE-LENGTH WALK

Random-walk-based network embedding algorithms, such as *Node2Vec* and *DeepWalk*, proposed a fixed walk length for the walks starting from every node. On the one hand, walks with long length may sample a lot of repeated nodes, such as back and forth walks between low-degree nodes. Node pairs generated by such walks may increase the risk of over-fitting. In addition, a fixed length for every walk may directly increase the sampling time as well

as storage and computation cost for learning node embeddings, and thus restrict the algorithm's scalability to large-scale networks. On the other hand, short-length walks lack ability to sufficiently capture the network topology such as community structure.

Considering the limitation of a fixed length for every walk, we propose a variable-length walk strategy based on the centrality of nodes. There are many ways to define node centrality in a network, such as degree [31], closeness [32], betweenness [33], PageRank and HITS. Betweenness centrality and closeness centrality involve calculating the shortest paths between all pairs of nodes on a network, which is unfeasible for large-scale networks with millions of nodes. Therefore, we simply ignore them and use the other three centrality metrics as in our variable-length strategy respectively. We use the hub value as the centrality value for every node in HITS algorithm.

We use degree centrality as an example to illustrate how we consider node centrality while setting the length of each walk. We propose walk starting from a high-degree source node to have a long length. In other words, for every source node u as the start of a walk, the length of the walk is give by

$$L(u) = \min\{d(u), L_{\max}\} + 1 \tag{5.6}$$

where $L(u)$ is the length of a walk started from node u, $L_{\max}$ is the upper bound of walk length. Algorithm 2 describes the complete high-degree biased variable-length random walk sampling strategy.

We use degree, PageRank and HITS as three centrality metrics while determining the walk length in the random walk sampling process. We perform these three different *DiaRW*s on BA networks with network sizes ranging from 10^3 to 10^5 to get embeddings for every node. The time of centrality computation and the AUC of link prediction task based on the corresponding node embeddings are shown in Table 5.2. The results show that that these three centrality measures achieve almost equal performances on link prediction task. However, PageRank and HITS require much more computation time compared to degree centrality. Therefore, we will focus on using degree centrality to determine walk length in the following analysis.

We compare the sampling storage size and time for high-degree biased random walk sampling strategy of *DiaRW* and uniform random walk sampling strategy of *DeepWalk* in Figure 5.4 on BA networks with network sizes ranging from 10^2 to 10^6. We set all the parameters to be the same except that in *DeepWalk*, the walk length of every walk is fixed as 80 whereas in our work, we set $L_{\max} = 80$. We find that the storage size of walks generated by high-degree biased random walk is lower than that from uniform random walk. High-degree biased random walk is able to finish sampling the BA network with millions of nodes in dozens of minutes while it takes several hours for uniform random walk with the same network size. We further show the degree distribution of node pair sets from walks generated by high-degree biased random walk on the BA network with size 2^{16} in Figure 5.5. The slope of the distribution is 2.66, which is much closer to the degree distribution of the original network (2.67 in Figure 5.1(a)). This indicates that high-degree biased random walk sampling strategy can better preserve network properties, such as degree distribution.

5.3.4. THE *DiaRW* ALGORITHM

The pseudo-code of our entire algorithm *DiaRW* is given in Algorithm 3. The algorithm consists of two main components: (1) a sampling generator and (2) a learning procedure. Algorithm 2 serves as the sampling generator, and we use algorithm 1 shown in 5.3.1 to train and learn the node representations.

Table 5.2: Evaluation of different centrality based *DiaRW* (link prediction AUC/computational time)

N	Degree	PageRank	HITS
1000	$0.61/5.4 \times 10^{-4}$	0.60/0.24	0.60/0.81
10000	$0.64/6.0 \times 10^{-3}$	0.64/2.24	0.64/13.42
100000	$0.64/6.0 \times 10^{-2}$	0.64/21.65	0.64/280.48

Algorithm 2: DiaRW_walk(G, u, $L_{\max}$)

Input: Network $G(V,E)$, max walk length $L_{\max}$
Output: Node sequence walks
1 Inititalize walks to $[u]$
2 $l = \min\{\mathrm{Deg}(u), L_{\max}\} + 1$
3 **for** $i = 0$ **to** l **do**
4 curr = walk$[-1]$
5 Select a node v uniformly from neighbors of curr
6 Append v to walks
7 Generate a random value $p \in [0,1]$
7 **if** $p < (1 - \frac{d(v)}{d(u)})$ **then**
9 Append curr to walks
10 **end for**
11 **return** walks

Algorithm 3: DiaRW(G, $L_{\max}$, k, w, d)

Input: Network $G(V,E)$
max walk length $L_{\max}$
walks per node k
window size w
embedding dimension d
Output: matrix of node representations $\Phi \in R^{|V| \times d}$
1 Initialize walks to empty
2 **for** iter = 1 **to** k **do**
3 **for** all nodes $u \in V$ **do**
4 walks = DiaRW_walk(G, u, $L_{\max}$)
5 Append walks to walks
6 SkipGram(Φ, walks, w)
7 **end for**
8 **end for**
9 **return** Φ

5.4. EXPERIMENTAL EVALUATION

We compare our *DiaRW* network embedding algorithm with four other baseline algorithms, i.e., *DeepWalk* [5], *Node2Vec* [6], *LINE* [26] and *HOPE* [27], on the tasks of multi-label node

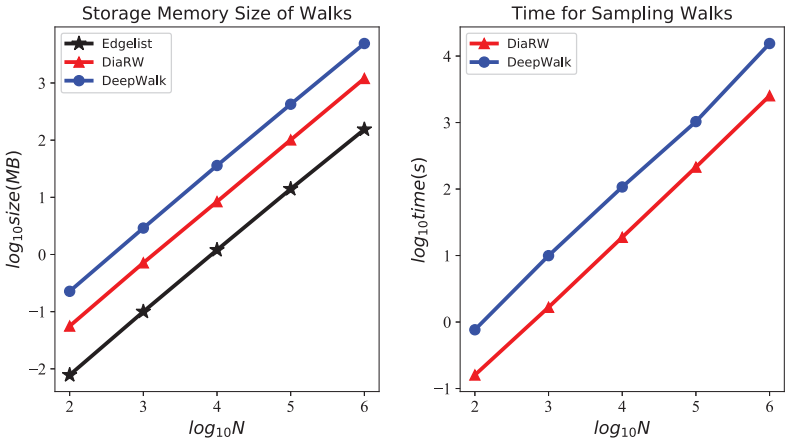


Figure 5.4: The Space and Time cost of sampling walks for random walk strategy proposed by *DiaRW* and *DeepWalk*. The X-axis is the number of nodes N in a network, Y-axes are the storage size of the sampled walks and time cost of the sampling process in the left and right panel of the figure, respectively. We show the size of BA network ranges from 100 to 1,000,000.

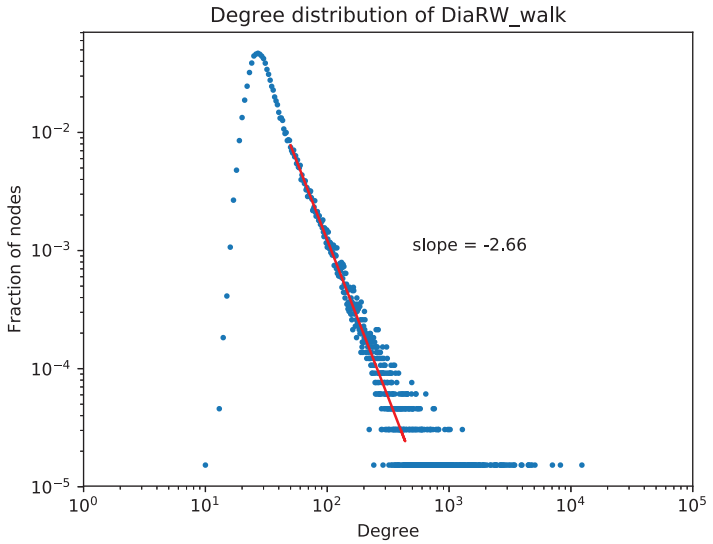


Figure 5.5: The node degree distribution of the node pair set generated by *DiaRW* random walk for BA graph with size 2^{16}.

classification and link prediction. Our experiment environment is listed in Table 5.3.

Table 5.3: Experiment environment.

OS	CentOS 4.8.5-16 Linux 4.4.114
MEMORY	128GB
DISK	300GB
CPU	Xeon(R) CPU E5-2620 v4 @ 2.10GHz

5.4.1. NETWORK DATASETS

Table 5.4: Network Datasets. Properties of the network datasets. The network size (N), the number of links ($|E|$), the average degree (Avg deg), link density (Density) and the average clustering coefficient (Avg cc) are shown. The first three network datasets are used for node classification task, we also show the number of labels in these three networks.

Network	N	$\|E\|$	Avg deg	Density	Avg cc	Labels
YouTube	1,134,890	2,987,624	5.27	4.6×10^{-6}	0.40	47
PPI	56,944	818,716	28.75	5.0×10^{-4}	0.18	121
Flickr	80,513	5,899,882	146.55	1.8×10^{-2}	0.16	194
Email-EU-core	1,005	25,571	33.24	3.0×10^{-2}	0.39	
Wiki-Vote	7,115	103,689	28.32	3.0×10^{-3}	0.14	
p2p-Gnutella	8,114	26,013	6.41	7.9×10^{-4}	7.2×10^{-3}	
Astroph	18,722	198,110	21.10	1.1×10^{-3}	0.63	
Cit-HepPh	34,546	421,578	24.36	7.0×10^{-4}	0.28	
Epinions	75,877	508,837	10.69	1.4×10^{-4}	0.13	
Twitter	11,316,811	85,331,846	11.23	9.9×10^{-7}	0.14	

Table 5.4 shows some properties of network datasets that will be used in our experiments. To show the efficiency and effectiveness of our algorithm, we choose the datasets of different sizes, ranging from thousands to millions of nodes. The detailed description of the network datasets for the multi-label classification task is given as follows:

• **YouTube** [34]: A social network between users on Youtube. It contains 1,157,827 nodes, 4,945,382 edges and 47 labels. The labels represent groups of users who enjoy common video genres.

• **Protein-Protein Interaction (PPI)** [35]: A network represents the interaction between proteins. It contains 56,944 nodes, 818,716 edges and 121 labels. Each of the labels corresponds to a biological function of the proteins.

• **Flickr** [34]: A network represents the contact between users in a photo sharing website. It contains 80,513 nodes, 5,899,882 unweighted edges and 194 labels. The labels represent the interest groups of the users, such as 'black and white photos'.

The network datasets that will be used for link prediction task are described as follows:

• **Email-EU-core** [36]: This is a network represent the email contact between users in a large European research institution. The emails only represent communication between institution members (the core), and the dataset does not contain incoming messages from or outgoing messages to the rest of the world. It contains 1,005 nodes and 25,571 edges.

• **Wikipedia vote network (Wiki-Vote)** [37]: The network extracted all administrator election and voting history data from Wikipedia community, where node represents the users who had participated in the election or been elected and edge indicates a voting process. The network contains 7,115 nodes and 103,689 edges.

• **Gnutella peer-to-peer network (p2p-Gnutella)** [36]: A sequence of snapshots of the Gnutella peer-to-peer file sharing network from August, 2002. Nodes represent hosts in the Gnutella network and edges represent connections between the them. The network contains 8,114 nodes and 26,013 edges.

• **High-energy physics citation network (Cit-HepPh)** [38]: This is a citation network generated from papers submitted to the e-print arXiv, where nodes represent papers. If a paper cites another paper, the network contains an edge between them. It has 34,546 nodes and 421,578 edges.

• **Astrophy collaboration** [36]: This is a collaboration network generated from papers submitted to the e-print arXiv, where nodes represent scientists, and an edge is formed between two scientists if they have collaborated on one paper. The network has 18,722 nodes and 198,110 edges.

• **Epinions** [39]: The network represents who-trust-whom relationships between users of the epinions.com product review website. It has 75,877 nodes and 508,837 edges.

• **YouTube** [34]: The same dataset used in the node-classification task.

• **Twitter** [40]: The nodes represent users and friends are represented using edges in Twitter. It contains 11,316,811 nodes and 85,331,846 edges.

5.4.2. BASELINE METHODS

We use the following four algorithms as the baselines:

• **DeepWalk** [5]: *DeepWalk* adopts uniform random walk to sample walks from a network and Skip-Gram model to generate network embeddings for each node.

• **Node2Vec** [6]: *Node2Vec* emploits a biased random walk to sample walks from a network. The biased random walk gives a trade-off between breadth-first-like sampling and depth-first-like sampling of the neighborhood. It contains the in-out and return hyper-parameters p and q. The learning model is Skip-Gram as well. We perform a grid search over $p, q \in \{0.25, 0.5, 1, 2, 4\}$ and 10-fold cross-validation to obtain the optimal embeddings, as suggested by [6].

• **LINE** [26]: *LINE* first learns the embedding vectors of nodes which preserve the first- and second-order proximities, respectively, and then concatenates them as the final embeddings.. We use the *LINE* (1st and 2nd) method which has shown the best performance in their paper. The original version of *LINE* is implemented in C++, for comparison fairness, we use an implementation of *LINE* in Python with TensorFlow.

• **HOPE** [27]: This method defines similarity measures between nodes which are useful for preserving higher-order proximity and formulates these measures as a product of sparse matrices to efficiently find the latent representations. The authors experimented with different similarity measures, including Katz Index, Rooted PageRank, Common Neighbors, and Adamic-Adar score. The Katz index with decay parameter $\beta = 0.1$ is selected for HOPE's high-order proximity measurement, since this setting gives the best performance in the original paper.

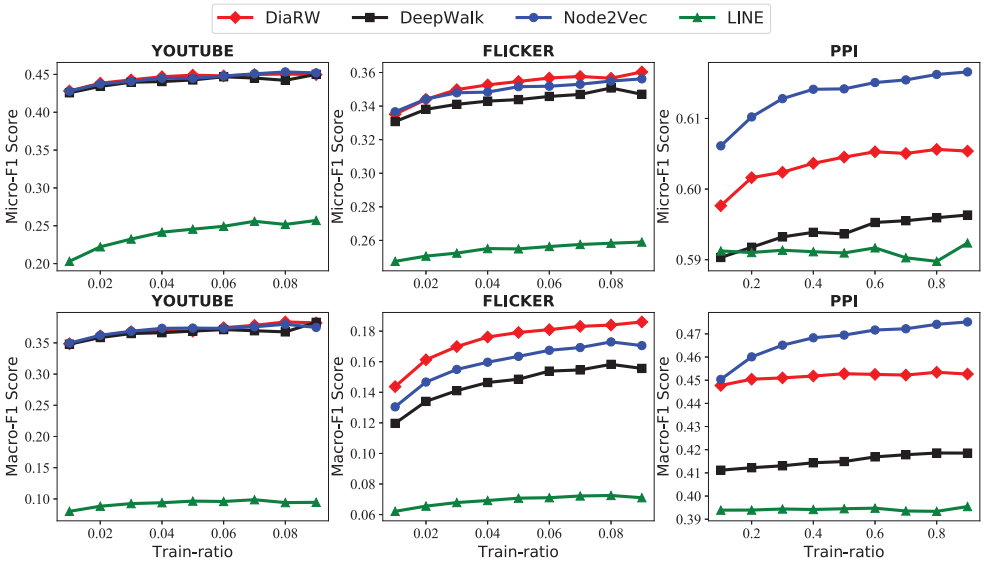


Figure 5.6: Performance evaluation on various datasets for multi-label classification.

Table 5.5: Time cost on embedding for multilabel classification (in seconds).

	DiaRW	*Node2Vec*	*DeepWalk*	*LINE*
YouTube	**3,391**	581,726	31,668	272,833
PPI	**290**	1,573	572	2,538
Flickr	**2,166**	179,917	2,487	11,001

5.4.3. EXPERIMENTS ON MULTI-LABEL CLASSIFICATION

Predicting node labels using network topology is widely applied in modern applications ranging from document classification [41] to interest prediction [42]. Among these applications, multi-label node classification is significantly challenging, especially for networks with a large number of labels. To perform this task, we use the learned node vector and an one-vs-rest logistic regression classifier (using the LIBLINEAR library with L2 regularization) [43]. When training the classifier, we randomly sample a portion of the labeled nodes as the training set and the rest as the test set. For PPI, we randomly sample 10% to 90% of the nodes as the training samples and use the remaining ones to test the performance. For Flicker and YouTube, we randomly sample 1% to 10% of the nodes as the training samples and use the left nodes to test the performance, which corresponds to the fact that these two datasets have only a small part of labeled nodes for entire networks. We repeat the experiment for 5 times and report the averaged Micro-F1 and Macro-F1.

The results are shown in Figure 5.6 and Table 5.5. Since *HOPE* fails to learn the embeddings in our current experimental environment for all the datasets on multi-label classification, we only show the results of the remaining three algorithms and compare them with *DiaRW*. The time cost for learning embeddings is given in Table 5.5. From Figure 5.6 and Table 5.5, we have the following observations and analysis:

• In Figure 5.6, we observe that random walk based algorithms, i.e., *DeepWalk*, *Node2Vec*

and *DiaRW*, outperform *LINE* in the multi-label classification task. The main reason can be inferred from the fact that *LINE* simply aims to capture low-order proximities for nodes, i.e., only nodes which are at most two hops away from a center node are considered as its neighbors. This is not enough for node classification as high-order proximity neighbors can also be classified by the same labels. In contrast, by generating random walks in the network, the neighborhoods are not restricted to just one-hop or two-hop neighbors.

• In Figure 5.6, *Node2Vec* performs relatively good in the multi-label classification task. *Node2Vec* performs better than or or at least similar to *DeepWalk* in all datasets, indicating biased random walks have better adaptability and accuracy for capturing network structures than uniform random walks. Despite the gain for accuracy over *DeepWalk*, *Node2Vec* is far less efficient than *DeepWalk* (Table 5.5). It takes at least three times longer than *DeepWalk* to learn the embeddings for the same dataset, which is even more for large-scale networks. This is because, *Node2Vec* requires a preprocess procedure to compute and store the interconnections between the neighbors of every node for second-order random walks, which is pretty expensive on both time and space for large-scale networks, therefore greatly affects the efficiency and scalability of embedding.

• *DiaRW* shows a competitive performance to *Node2Vec*, but with much higher efficiency. Specifically, regarding to the Macro-F1 and Micro-F1 score, *DiaRW* shows comparable results as *Node2vec* and *DeepWalk* in YouTube network. In PPI network, *DiaRW* improves Macro-F1 score by 9.8% and Micro-F1 score by 1.5% over *DeepWalk*. In Flicker network, *DiaRW* outperforms all the baselines, gaining 12.5% improvement on Macro-F1 score compared to *Node2vec*. Taking time cost showing in Table 5.5 together, we can conclude that *DiaRW* can finish embedding multiple times faster than *DeepWalk* and dozens of times faster than *Node2Vec* while maintaining the accuracy for multi-label classification. The huge gains in time are mainly due to the variable-length walk strategy we adopt, which has drastically reduced the size of walks, and accelerated walking and training as well.

• The experimental results imply that there is a lot of redundant node pairs generated from walks sampled by uniform random walk with fixed walk length, which not only is useless for improving accuracy but also greatly slow down the algorithm. For networks with many types of labels but short of labeled data such as Flicker [34], *DiaRW* gets even better performance than state-of-the-art algorithms due to better representation of network structure.

5.4.4. EXPERIMENTS ON LINK PREDICTION

Networks are constructed from the observed links between nodes, which may be incomplete or inaccurate. The challenge often lies in identifying spurious interactions and predicting missing links. Link prediction refers to the task of predicting either missing links in a static network or links that may appear in the future in temporal networks [11, 44]. When we represent network nodes as embedding vectors, the dot product of two embedding vectors of nodes u and v is treated as the similarity score $S(u, v)$ between the two nodes. Nodes with higher similarity score are considered to be more likely to connect.

To perform link prediction in a network, we first randomly remove half of its edges. The node representations are then learned from the remaining part of the network, i.e., training data. To create negative labels for the prediction task, we randomly select pairs of nodes that are not connected in the original network. The number of such pairs is equal to the number of removed edges. The "negative" pairs and the pairs from edges that have been removed, are used together to form the labeled data as the test set. Given embedding vectors $\Phi(u)$ and $\Phi(v)$ of two nodes u and v, we define the similarity score $S(u, v)$ as dot product $\Phi(u) \cdot \Phi(v)$. We use

Table 5.6: Area Under Curve (AUC) scores for link prediction.

	DiaRW	*Node2Vec*	*DeepWalk*	*HOPE*	*LINE*
Email-EU-core	0.8572	0.8222	0.8270	**0.8618**	0.7870
Wiki-Vote	**0.9356**	0.7957	0.7946	0.9283	0.8101
p2p-Gnutella	**0.7641**	0.7025	0.6947	0.6254	0.6878
Cit-HepPh	0.9517	**0.9576**	0.9472	0.5125	0.6878
Astrophy	0.9159	**0.9217**	0.9088	0.5320	0.8896
Epinions	**0.8972**	0.8512	0.8463	*	0.8417
YouTube	**0.7985**	0.7726	0.7681	*	0.6463
Twitter	**0.9010**	*	*	*	*

Table 5.7: Time cost for link prediction task (seconds).

	DiaRW	*Node2Vec*	*DeepWalk*	*HOPE*	*LINE*
Email-EU-core	3	25	8	**1**	10
Wiki-Vote	**10**	113	28	180	29
p2p-Gnutella	**7**	111	59	25	87
Cit-HepPh	**53**	405	167	725	230
Astrophy	**46**	438	175	157	953
Epinions	**109**	3,466	1,069	*	4,066
YouTube	**1,448**	581,726	31,668	*	272,833
Twitter	**37,167**	*	*	*	*

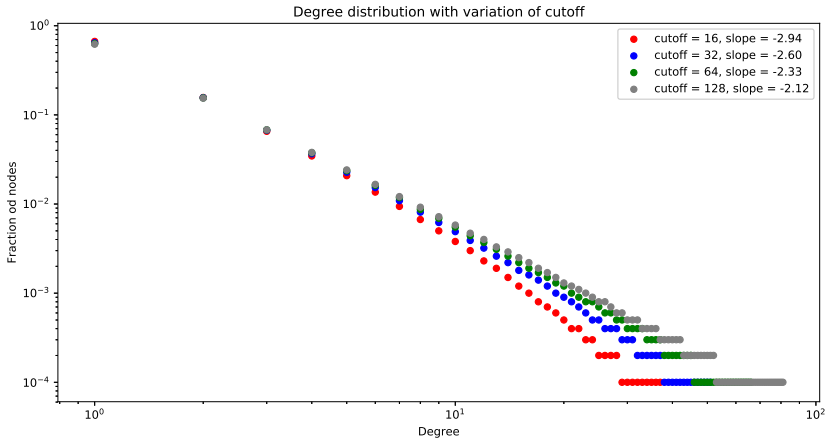


Figure 5.7: Degree distribution of synthetic networks with different cutoff κ.

Area Under Curve (AUC) to evaluate the performance of the network embedding algorithms on link prediction task. We repeated the splitting of the network data into training and test set for 10 times. Then we run the network embedding algorithm and link prediction task on each of the 10 splits to get 10 AUC scores. The AUC scores show in the following studies are all averaged over 10 times. The results are given in Table 5.6. We show the time cost

Table 5.8: The impact of degree heterogeneity on the performance of link prediction task.

Cutoff κ	$\|V\|$	$\|E\|$	CV	Hm	AUC of *DiaRW*	AUC of *DeepWalk*
16	10,000	10,252	1.26	0.06	0.5059	0.4724
32	10,000	11,329	1.53	0.07	0.5692	0.4865
64	10,000	12,714	1.78	0.07	0.6279	0.4956
128	10,000	14,007	2.10	0.08	0.6616	0.5136

of each embedding algorithm performed on every network dataset in Table 5.7, where "*" means the algorithm fails under the limitation of computation resources (Table 5.3) and time requirement (one week). From the results, we have the following observations and analysis:

• *DiaRW* outperforms almost all the other algorithms on link prediction task. More precisely, it achieves 8.8% improvement on AUC for p2p-Gnutella, 5.4% for Epinions and 3.95% for Youtube, compared to the best AUC from baseline algorithms. For the two smaller datasets, i.e., Email-EU-core and Wiki-Vote, *DiaRW* shows competitive performance to *HOPE* on Email-EU-core, and it performs as good as *Node2Vec* for Cit-HepPh and Astrophy networks. In a word, experimental results sufficiently show the advantage of *DiaRW* on link prediction task, which can well adapt to networks of various sizes and types.

• *LINE* shows comparable performance to random-walk-based algorithms on several datasets. This suggests that low-order proximity is useful for predicting missing links. However, real-world networks tend to be so sparse that it's hard to sample enough low-order proximities for representation learning. In view of this, random-walk-based methods is more flexible and effective as they use a random walk to enrich the neighbors of nodes, which is able to introduce higher-order proximities. The performance of *HOPE* is highly dependent on the dataset, which implies its poor adaptability to different networks.

• Results from Table 5.7 imply once again that our method is scalable and efficient for large-scale networks. Taking Twitter as an example, all the algorithms except *DiaRW* fail to obtain the node representations. In contrast, it takes only ten hours for *DiaRW* to learn the embeddings for Twitter, with a superior performance on link prediction task.

We further explore the impact of degree heterogeneity on the performance of embedding. We use the model proposed in [45] to generate synthetic networks with different degree heterogeneity by multiplying the parameter exponential cutoff κ from 32 to 128 with $\alpha = 2$. All the synthetic network has the same network size. The degree distributions of these synthetic networks are shown in Figure 5.7. The degree heterogeneity of the networks are evaluated by coefficient of variation(CV), defined as the ratio of the standard deviation to the mean of the degree sequence, and a degree heterogeneity measure(Hm) proposed in [46]. Higher values of both CV and Hm indicate the network is more heterogeneous regarding to node degree. We also compute the AUC on link prediction task to evaluate the embeddings learned by *DiaRW* and *DeepWalk*, respectively. The results are given in Table 5.8. We find that, with the increase of degree heterogeneity, *DiaRW* achieves significant improvement on AUC. Additionally, *DiaRW* shows great advantages over *DeepWalk*, providing evidence that *DiaRW* can better adapt to degree heterogeneity property of real-world networks than uniform random walks.

5.4.5. SEPARATE EFFECT OF BACKTRACKING AND VARIABLE-LENGTH

Our walk strategy *DiaRW_walk* can be divided into two sub-strategies, i.e., high-degree biased backtracking and variable-length walk. In order to explore their effects separately, we

design two variants of *DiaRW* based on these two sub-strategies, named as *DiaRW_BT* and *DiaRW_VarL*, respectively. Taking several networks from Section 5.4.1 as examples, we use link prediction task to evaluate *DiaRW_BT* and *DiaRW_VarL*, along with *DiaRW* and *DeepWalk* as comparisons. As shown in Table 5.9, *DiaRW_VarL* performs similar as *DeepWalk*, because it aims to improve the efficiency of network embedding. While *DiaRW_BT* indeed increase the accuracy on link prediction task. Also, this improvement can be further improved when combining with variable-length walk, verifying the illustration from Section 5.3.3 that high-degree nodes need larger walk length to cover the loss of frequent backtracking. In summary, we conclude that both of the two sub-strategies contribute to the network embedding, in which the high-degree biased backtracking directly improve the accuracy of node representations, and variable-length walk gains huge improvement on efficiency.

Table 5.9: Evaluations of *DiaRW_BT* and *DiaRW_VarL* on link prediction task. We use AUC to quantify the performance.

	DeepWalk	*DiaRW_BT*	*DiaRW_VarL*	*DiaRW*
Email-EU-core	0.8270	0.8475	0.8229	0.8572
Wiki-Vote	0.7946	0.8477	0.7922	0.9356
Cit-HepPh	0.9472	0.9543	0.9336	0.9517
Astrophy	0.9088	0.9151	0.8885	0.9159
Epinions	0.8463	0.8673	0.8188	0.8852

5.4.6. PARAMETER SENSITIVITY

We explore how the different values of parameters affect the performance of *DiaRW*. Figure 5.8 shows the AUC gained by *DiaRW* on link prediction task for p2p-Gnutella network. Except for the parameter being tested, all the other parameters in the experiment are set to their default value. We find that the parameters related to the random walk process, i.e., the number of walks starting from per node k, maximum walk length $L_{\max}$, have impact on the performance of embedding. The increase of $L_{\max}$ can increase the AUC score for link prediction until $L_{\max} = 90$. Also, we get the optimal AUC when $k = 10$. Similarly, we observe that increasing the context size w for Skip-Gram model will also improve the AUC since larger w could discovery higher order relationships in the network which is helpful for preserving network proximity. However, once this parameter is set too large, it will in turn introduce noise, attenuating the impact of closer neighborhoods, which accounts for the slight degradation of performance. We further test how the dimension d of a embedding vector would affect the AUC score.We infer that vectors with too small dimensions lack expressive ability, embedding in this representation space may not be able to preserve the structure information of the networks, whereas continuously increasing the number of vector dimensions by adding more nodes on the hidden layer of neural network will increase the risk of over-fitting problem, which could also negatively affect the performance.

5.4.7. SCALABILITY

To test scalability of our algorithm, we embed BA networks with increasing node size from 2^{10} to 2^{20} nodes . Figure 5.9 depicts the time cost required for sampling walks and the total time cost for both sampling walks and learning process. *DiaRW* is able to learn embeddings for networks with millions of nodes in dozens of hours. The time difference between the sampling and learning process indicates that the main time cost comes from the sampling

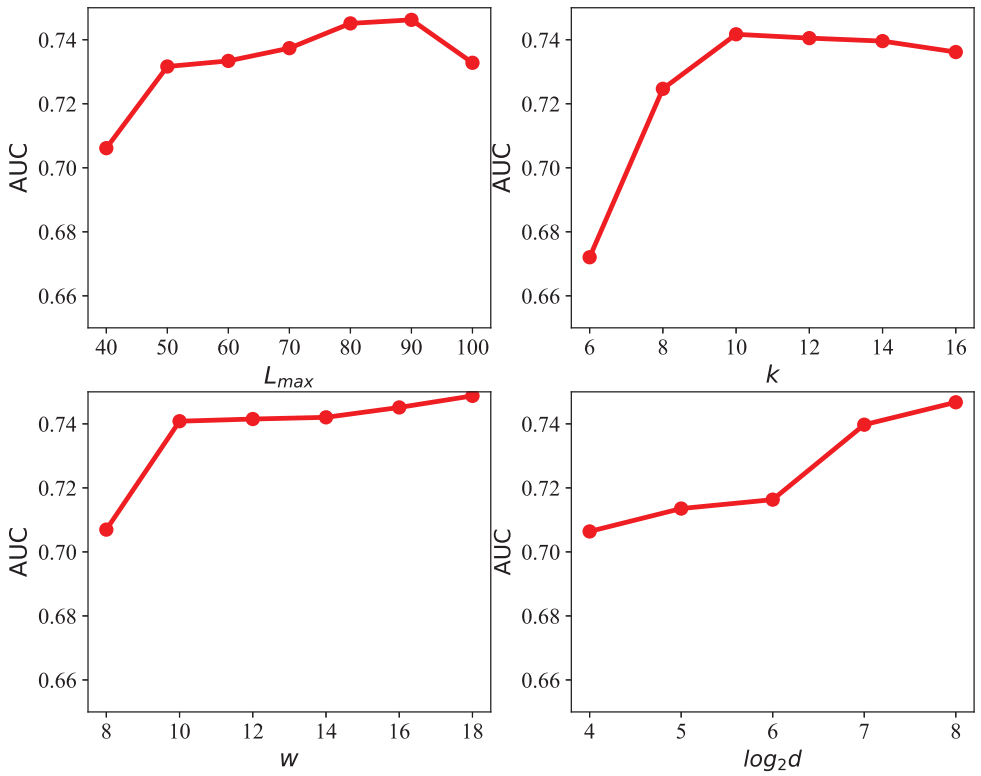


Figure 5.8: Parameter sensitivity of *DiaRW* in p2p-Gnutella for link prediction task. We show how the AUC changes with the max length of the walk ($L_{\max}$), the number of walks per node (k), the context size (w) and the dimension of the embedding vector (d). For the dimension value, we show $log_2 d$ in the x-axis.

process. This further emphasize the importance of designing a efficient sampling strategy for network embedding algorithm.

5.5. CONCLUSION

In this work, we proposed *DiaRW*, an efficient method for network embedding, which can easily scale to networks with millions of nodes and billions of edges. The core of *DiaRW* is the sampling strategy based on biased backtracking mechanism and variable-length mechanism. *DiaRW* can well adapt the degree heterogeneity of real-world networks and and greatly reduce the redundant node pairs as input for Skip-Gram when compared with fixed-length random walks. Compared to the state-of-the-art baseline algorithms, it shows higher performance and efficiency. As future work, we plan to extend our algorithm to networks with special properties such as heterogeneous information networks, networks with explicit domain features for nodes and edges, and signed networks.

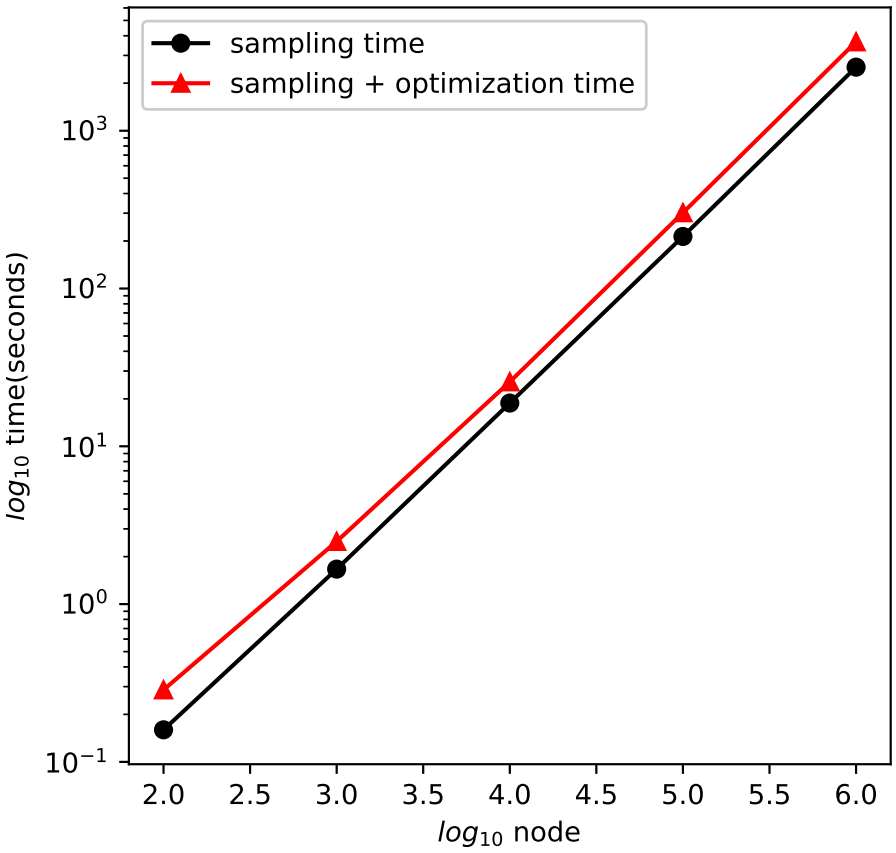


Figure 5.9: Scalability of *DiaRW* on BA networks. We show the network size ranges from 100 to 1,000,000. The x-axis is the network size, the y-axis is the time cost of the embedding algorithm *DiaRW*.

REFERENCES

[1] M. Eirinaki, J. Gao, I. Varlamis, and K. Tserpes, *Recommender systems for large-scale social networks: A review of challenges and solutions,* Future Generation Computer Systems **78**, 413 (2018).

[2] S. D. Cardoso, F. K. Amanqui, K. J. A. Serique, J. L. C. dos Santos, and D. A. Moreira, *Swi: A semantic web interactive gazetteer to support linked open data,* Future Generation Computer Systems **54**, 389 (2016).

[3] J. Kamruzzaman, G. Wang, G. Karmakar, I. Ahmad, and M. Z. A. Bhuiyan, *Acoustic sensor networks in the internet of things applications,* Future Generation Computer Systems **86**, 1167 (2018).

[4] A. Theocharidis, S. van Dongen, A. J. Enright, and T. C. Freeman, *Network visualization and analysis of gene expression data using biolayout express(3d),* Nature protocols **4**, 1535 (2009).

[5] B. Perozzi, R. Al-Rfou, and S. Skiena, *Deepwalk: Online learning of social representations,* in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD'14 (ACM, New York, NY, USA, 2014) pp. 701–710.

[6] A. Grover and J. Leskovec, *Node2vec: Scalable feature learning for networks,* in *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD'16 (ACM, New York, NY, USA, 2016) pp. 855–864.

[7] S. Bhagat, G. Cormode, and S. Muthukrishnan, *Node classification in social networks,* Computer Science **16**, 115 (2011).

[8] A. Y. Ng, M. I. Jordan, and Y. Weiss, *On spectral clustering: Analysis and an algorithm,* in *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, NIPS'01 (MIT Press, Cambridge, MA, USA, 2001) pp. 849–856.

[9] A. Faroughi and R. Javidan, *Canf: Clustering and anomaly detection method using nearest and farthest neighbor,* Future Generation Computer Systems **89**, 166 (2018).

[10] D. Liben-Nowell and J. Kleinberg, *The link prediction problem for social networks,* in *Proceedings of the Twelfth International Conference on Information and Knowledge Management*, CIKM'03 (ACM, New York, NY, USA, 2003) pp. 556–559.

[11] S. Aslan and M. Kaya, *Topic recommendation for authors as a link prediction problem,* Future Generation Computer Systems **89**, 249 (2018).

[12] P. E. Rauber, A. X. Falcão, and A. C. Telea, *Visualizing time-dependent data using dynamic t-sne,* in *Proceedings of the Eurographics / IEEE VGTC Conference on Visualization: Short Papers*, EuroVis'16 (Eurographics Association, Goslar Germany, Germany, 2016) pp. 73–77.

[13] J. Gómez-Romero, M. Molina-Solana, A. Oehmichen, and Y. Guo, *Visualizing large knowledge graphs: A performance analysis,* Future Generation Computer Systems **89**, 224 (2018).

[14] D. Luo, C. Ding, F. Nie, and H. Huang, *Cauchy graph embedding,* in *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11 (Omnipress, USA, 2011) pp. 553–560.

[15] S. A. Myers, A. Sharma, P. Gupta, and J. Lin, *Information network or social network: The structure of the twitter follow graph,* in *Proceedings of the 23rd International Conference on World Wide Web*, WWW'14 (ACM, New York, NY, USA, 2014) pp. 493–498.

[16] T. Mikolov, K. Chen, G. Corrado, and J. Dean, *Efficient estimation of word representations in vector space,* CoRR **abs/1301.3781** (2013).

[17] A. L. Barabási and R. Albert, *Emergence of scaling in random networks,* Science **286**, 509 (1999).

[18] W. Yuan, K. He, G. Han, D. Guan, and A. M. Khattak, *User behavior prediction via heterogeneous information preserving network embedding,* Future Generation Computer Systems **92**, 52 (2019).

[19] M. E. Mugavin, *Multidimensional scaling: a brief overview,* Nursing research **57**, 64 (2008).

[20] A. Nedich and A. Ozdaglar, *A geometric framework for nonconvex optimization duality using augmented lagrangian functions,* Journal of Global Optimization **40**, 545 (2008).

[21] B. Schölkopf, A. Smola, and K.-R. Müller, *Nonlinear component analysis as a kernel eigenvalue problem,* Neural Comput. **10**, 1299 (1998).

[22] S. Yan, D. Xu, B. Zhang, and H. J. Zhang, *Graph embedding: a general framework for dimensionality reduction,* in *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR'05, Vol. 2 (2005) pp. 830–837.

[23] S. T. Roweis and L. K. Saul, *Nonlinear dimensionality reduction by locally linear embedding,* Science **290**, 2323 (2000).

[24] M. Wang, W. Fu, S. Hao, H. Liu, and X. Wu, *Learning on big graph: Label inference and regularization with anchor hierarchy,* IEEE Transactions on Knowledge and Data Engineering **29**, 1101 (2017).

[25] M. Wang, W. Fu, S. Hao, D. Tao, and X. Wu, *Scalable semi-supervised learning by efficient anchor graph regularization,* IEEE Transactions on Knowledge and Data Engineering **28**, 1864 (2016).

[26] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, *Line:large-scale information network embedding,* in *Proceedings of the 24th International Conference on World Wide Web*, WWW'15 (International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 2015) pp. 1067–1077.

[27] M. Ou, P. Cui, J. Pei, Z. Zhang, and W. Zhu, *Asymmetric transitivity preserving graph embedding,* in *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD'16 (ACM, New York, NY, USA, 2016) pp. 1105–1114.

[28] Z.-K. Zhang, C. Liu, X.-X. Zhan, X. Lu, C.-X. Zhang, and Y.-C. Zhang, *Dynamics of information diffusion and its applications on complex networks,* Physics reports **651**, 1 (2016).

[29] C. Liu, X.-X. Zhan, Z.-K. Zhang, G.-Q. Sun, and P. M. Hui, *How events determine spreading patterns: information transmission via internal and external influences on social networks,* New Journal of Physics **17**, 113045 (2015).

[30] R. Cohen, S. Havlin, and D. Ben-Avraham, *Efficient immunization strategies for computer networks and populations,* Physical review letters **91**, 247901 (2003).

[31] L. C. Freeman, *Centrality in social networks conceptual clarification,* Social Networks **1**, 215 (1978).

[32] G. Sabidussi, *The centrality index of a graph,* Psychometrika **31**, 581 (1966).

[33] L. Freeman, *A set of measures of centrality based on betweenness,* Sociometry **40**, 35 (1977).

[34] L. Tang and H. Liu, *Scalable learning of collective behavior based on sparse social dimensions,* in *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM'09 (ACM, New York, NY, USA, 2009) pp. 1107–1116.

[35] M. Livstone, B.-J. Breitkreutz, C. Stark, L. Boucher, A. Chatr-Aryamontri, R. Oughtred, J. Nixon, T. Reguly, J. Rust, A. Winter, K. Dolinski, and M. Tyers, *The biogrid interaction database,* Nature Precedings **41**, : D637 (2011).

[36] J. Leskovec, J. Kleinberg, and C. Faloutsos, *Graph evolution: Densification and shrinking diameters,* ACM transactions on Knowledge Discovery from Data (TKDD) **1**, 2 (2007).

[37] J. Leskovec, D. Huttenlocher, and J. Kleinberg, *Signed networks in social media,* in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10 (ACM, New York, NY, USA, 2010) pp. 1361–1370.

[38] J. Gehrke, P. Ginsparg, and J. Kleinberg, *Overview of the 2003 kdd cup,* ACM SIGKDD Explorations Newsletter **5**, 149 (2003).

[39] M. Richardson, R. Agrawal, and P. Domingos, *Trust management for the semantic web,* in *The Semantic Web - ISWC 2003*, Vol. 2870 (2003) pp. 351–368.

[40] R. Zafarani and H. Liu, *Social Computing Data Repository at ASU* (Arizona State University, School of Computing, Informatics and Decision Systems Engineering, 2009).

[41] S. N. B. Bhushan and A. Danti, *Classification of compressed and uncompressed text documents,* Future Generation Computer Systems **88**, 614 (2018).

[42] B. Xu and H. Zhuge, *An angle-based interest model for text recommendation,* Future Generation Computer Systems **64**, 211 (2016).

[43] R. E. Fan, K. W. Chang, C. J. Hsieh, X. R. Wang, and C. J. Lin, *Liblinear: A library for large linear classification,* J. Mach. Learn. Res. **9**, 1871 (2008).

[44] X.-X. Zhan, A. Hanjalic, and H. Wang, *Information diffusion backbones in temporal networks,* arXiv preprint arXiv:1804.09483 (2018).

[45] M. E. Newman, *Spread of epidemic disease on networks,* Physical review. E, Statistical, nonlinear, and soft matter physics **66**, 16128 (2002).

[46] R. Jacob, K. P. Harikrishnan, R. Misra, and G. Ambika, *Measure for degree heterogeneity in complex networks and its application to recurrence network analysis,* Royal Society open science **4**, 160757 (2017).

6

SI-SPREADING-BASED NETWORK EMBEDDING IN STATIC AND TEMPORAL NETWORKS

This chapter have been submitted to EPJ Data Science as: X.-X. Zhan, Z. Li, N. Masuda, P. Holme and H. Wang, SI-spreading-based Network Embedding in Static and Temporal Networks

Link prediction can be used to extract missing information, identify spurious interactions as well as forecast network evolution. Network embedding is a methodology to assign coordinates to nodes in a low-dimensional vector space. By embedding nodes into vectors, the link prediction problem can be converted into a similarity comparison task. Nodes with similar embedding vectors are more likely to be connected.

Classic network embedding algorithms are random-walk-based. They sample trajectory paths via random walks and generate node pairs from the trajectory paths. The node pair set is further used as the input for a Skip-Gram model, a representative language model that embeds nodes (which are regarded as words) into vectors. In the present study, we propose to replace random walk processes by a spreading process, namely the susceptible-infected (SI) model, to sample paths. Specifically, we propose two SI-spreading-based algorithms, SINE and TSINE, to embed static and temporal networks, respectively. The performance of our algorithms is evaluated by the missing link prediction task in comparison with state-of-the-art static and temporal network embedding algorithms. Results show that SINE and TSINE outperform the baselines across all six empirical datasets. We further find that the performance of SINE is mostly better than TSINE, suggesting that temporal information does not necessarily improve the embedding for missing link prediction. Moreover, we study the effect of the sampling size, quantified as the total length of the trajectory paths, on the performance of the embedding algorithms. The better performance of SINE and TSINE requires a smaller sampling size in comparison with the baseline algorithms. Hence, SI-spreading-based embedding tends to be more applicable to large-scale networks.

6.1. INTRODUCTION

Real-world systems can be represented as networks, with nodes representing the components and links representing the connections between them [1, 2]. The study of complex networks pervades in different fields [3]. For example, with biological or chemical networks, scientists study interactions between proteins or chemicals to discover new drugs [4, 5]. With social networks, researchers tend to classify or cluster users into groups or communities, which is useful for many tasks, such as advertising, search and recommendation [6, 7]. With communication networks, learning the network structure can help understand how information spreads over the networks [2]. These are only a few examples of the important role of analyzing networks. For all these examples, the data may be incomplete. If so, it could be important to be able to predict the link most likely to be missing. If the network is evolving, it could be crucial to forecast the next link to be added. For both of these applications one needs link prediction [8–12].

In link prediction, one estimates the likelihood that two nodes are adjacent to each other based on the observed network structure [13]. Methods using similarity-based metrics, maximum likelihood algorithms and probabilistic models are major families of link prediction methods [14]. Recently, network embedding, which embeds nodes into a low-dimensional vector space, has attracted much attention in solving the link prediction problem [14, 15]. The similarity between the embedding vectors of two nodes is used to evaluate whether they would be connected or not. Different algorithms have been proposed to obtain network embedding vectors. A simplest embedding method is to take the row or column vector in the adjacency matrix, which is called an adjacency vector of the corresponding node, as the embedding vector. Then, the representation space is N-dimensional, where N is the number of nodes. As real-world networks are mostly large and sparse, the adjacency vector of a node

is sparse and high-dimensional. In addition, the adjacency matrix only contains the first-order neighborhood information, and therefore the adjacency vector neglects the high-order structure of the network such as paths longer than an edge. These factors limit the precision of network embedding based on the adjacency vector in link prediction tasks. Work in the early 2000s attempted to embed nodes into a low dimension space using dimension reduction techniques [16–18]. Isomap [16], locally linear embedding (LLE) [17] and Laplacian eigenmap [18] are algorithms based on the k-nearest graph, where nodes i and j are connected by a link in the k-nearest graph if the length of the shortest path between i and j is within the k-th shortest among the length of all the shortest paths from i to any other nodes. Matrix factorization algorithms decompose the adjacency matrix into the product of two low-dimensional rectangular matrices. The columns of the rectangular matrices are the embedding vectors for nodes. Singular value decomposition (SVD) [19] is one commonly used and simple matrix factorization. However, the computation complexity of most of the aforementioned algorithms is at least quadratic in terms of N, limiting their applicability to large networks with millions of nodes.

Random-walk-based network embedding is a promising family of computationally efficient algorithms. These algorithms exploit truncated random walks to capture the proximity between nodes [20–22] generally via the following three steps [23–25]: (1) Sample the network by running random walks to generate trajectory paths. (2) Generate a node pair set from the trajectory paths: each node on the trajectory path is viewed as a center node, the nearby nodes within a given distance are considered as the neighboring nodes. A node pair in the node pair set is formed by a center node and each of its neighboring nodes. (3) Apply a word embedding model such as Skip-Gram to learn the embedding vector for each node by using the node pair set as input. Skip-Gram assumes nodes that are similar in topology or content tend to have similar representations [22]. Algorithms have been designed using different random walks to capture high-order structure on networks. For example, DeepWalk [20] and Node2Vec [23] adopted uniform and biased random walks, respectively, to sample the network structure. In addition, random-walk-based embedding methods have also been developed for temporal networks, signed networks and multilayer networks [26–29].

In contrast to random-walk-based embedding, here we propose SI-spreading-based network embedding algorithms for static and temporal networks. We deploy the susceptible-infected (SI) spreading process on the given network, either static or temporal, and use the corresponding spreading trajectories to generate the node pair set, which is fed to the Skip-Gram to derive the embedding vectors. The trajectories of an SI spreading process capture the tree-like sub-network centered at the seed node, whereas random walk explores long walks that possibly revisit the same node. We evaluate our static network embedding algorithm, which refer to as *SINE*, and temporal network embedding, *TSINE*, via a missing link prediction task in six real-world social networks. We compare our algorithms with state-of-the-art static and temporal network embedding methods. We show that both *SINE* and *TSINE* outperform other static and temporal network embedding algorithms, respectively. In most cases, the static network embedding, *SINE*, performs better than *TSINE*, which additionally uses temporal network information. In addition, we evaluate the efficiency of SI-spreading-based network embedding via exploring the sampling size for the Skip-Gram, quantified as the sum of the length of the trajectory paths, in relation to its performance on the link prediction task. We show that high performance of SI-spreading-based network embedding algorithms requires a significantly smaller sampling size compared to random-walk-based embeddings. We further explore what kind of links can be better predicted to further explain why our

proposed algorithms show better performance than the baselines.

The rest of the chapter is organized as follows. We propose our method in Section 6.2. In Section 6.2.1, we propose our SI-spreading-based sampling method for static networks and the generation of the node pair set from the trajectory paths. Skip-Gram model is introduced in Section 6.2.2. We introduce an SI-spreading-based sampling method for temporal networks in Section 6.2.3. In Section 6.3, our embedding algorithms are evaluated on a missing link prediction task on real-world static and temporal social networks. The chapter is concluded in Section 6.4.

6.2. SI-SPREADING-BASED EMBEDDING

This section introduces SI-spreading-based network embedding methods. Firstly, we illustrate our SI-spreading-based network embedding method for static networks in Sections 6.2.1 and 6.2.2. Section 6.2.3 generalizes the method to temporal network embedding.

Because we propose the network embedding methods for both static and temporal networks, we start with the notations for temporal networks, of which the static networks are special cases. A temporal network is represented as $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ is the node set and $\mathcal{L} = \{l(i, j, t), t \in [0, T], i, j \in \mathcal{N}\}$ is the set of time-stamped contacts. The element $l(i, j, t)$ in $\mathcal{L}$ represents a bidirectional contact between nodes i and j at time t. We consider discrete time and assume that all contacts have a duration of one discrete time step. We use $[0, T]$ to represent the observation time window, $N = |\mathcal{N}|$ is the number of nodes. The aggregated static network $G = (\mathcal{N}, E)$ is derived from a temporal network $\mathcal{G}$. Two nodes are connected in G if there is at least one contact between them in $\mathcal{G}$. E is the edge set of G. The network embedding problem is formulated as follows:

Given a network $G = (\mathcal{N}, E)$, static network embedding aims to learn a low-dimensional representation for each node $i \in \mathcal{N}$. The node embedding matrix for all the nodes is given by $\mathbf{U} \in R^{d \times N}$, where d is the dimension of the embedding vector ($d < N$). The i-th column of $\mathbf{U}$, i.e., $\overrightarrow{u_i} \in R^{d \times 1}$, represents the embedding vector of node i.

6.2.1. SI-SPREADING-BASED STATIC NETWORK SAMPLING

The SI spreading process on a static network is defined as follows: each node is in one of the two states at any time step, i.e., susceptible (S) or infected (I); initially, one seed node is infected; an infected node independently infects each of its susceptible neighbors with an infection probability β at each time step; the process stops when no node can be infected further. To derive the node pair set as the input for Skip-Gram, we carry out the following steps:

CONSTRUCTION OF SPREADING TRAJECTORY PATHS.

In each run of the SI spreading process, a node i is selected uniformly at random as the seed. The SI spreading process starting from i is performed. The spreading trajectory $\mathcal{T}_i(\beta)$ is the union of all the nodes that finally get infected supplied with all the links that have transmitted infection between node pairs.

From each of the spreading trajectory $\mathcal{T}_i(\beta)$, we construct m_i trajectory paths, each of which is the path between the root node i and a randomly selected leaf node in $\mathcal{T}_i(\beta)$. The number m_i of trajectory paths to be extracted from $\mathcal{T}_i(\beta)$ is assumed to be given by

$$m_i = \max\left\{1, \frac{\mathcal{K}(i)}{\sum_{j \in \mathcal{N}} \mathcal{K}(j)} m_{\max}\right\},$$

Algorithm 1 Generation of trajectory paths from SI spreading

Input: $G = (\mathcal{N}, E)$, B, $L_{\max}$, β, m_i
Output: node trajectory path set D

Initialize number of context windows $C = 0$
Initialize node trajectory path set $D = \varnothing$
while $B - C > 0$ **do**
    Randomly choose node i as the seed to start the SI spreading
    Generate spreading trajectory tree $\mathcal{T}_i(\beta)$
    Randomly choose m_i trajectory paths $D_{g_i}(g_i = 1, \ldots, m_i)$ from $\mathcal{T}_i(\beta)$
    for $g_i = 1, \ldots, m_i$ **do**
        if $|D_{g_i}| > L_{\max}$ **then**
            Choose the first $L_{\max}$ nodes from D_{g_i} to form $D^*_{g_i}$
            Add the trajectory $D^*_{g_i}$ to D
            $C = C + |D^*_{g_i}|$
        else
            Add the trajectory D_{g_i} to D
            $C = C + |D_{g_i}|$
        end if
    end for
end while
return D

where $m_{\max}$ is a control parameter and $\mathcal{K}(i)$ is the degree of the root node i in the static network (or aggregated network).

The trajectory paths may have different lengths (i.e., number of nodes in the path). For a trajectory path whose length is larger than $L_{\max} = 20$, we only take the first $L_{\max}$ nodes on the path. For a randomly chosen seed node i, we can generate m_i trajectory paths from $\mathcal{T}_i(\beta)$. We stop running the SI spreading process until the sum of the length of the trajectory paths reaches the sampling size $B = NX$, where X is a control parameter. We consider $X \in \{1, 2, 5, 10, 25, 50, 100, 150, 200, 250, 300, 350\}$. We compare different algorithms using the same B for fair comparison [26] to understand the influence of the sampling size. We show how to sample the trajectory paths in Algorithm 1.

NODE PAIR SET GENERATION.
We illustrate how to generate the node pairs, the input of the Skip-Gram, from a trajectory path in Figure 6.1. Consider a trajectory path, $1, 3, 6, 8, 9, 10, 7, 5$, starting from node 1 and ending at node 5. We set each node, e.g., node 3, as the center node, and the neighboring nodes of the center node are defined as nodes within $\omega = 2$ hops. The neighboring nodes of node 3 are, 1, 6 and 8. We thus obtain ordered node pairs $(3, 1)$, $(3, 6)$, and $(3, 8)$. Thus, we use the union of node pairs centered at each node in each of trajectory path as the input to the Skip-Gram model.

6.2.2. SKIP-GRAM MODEL

We illustrate how the Skip-Gram derives the embedding vector for each node based on the input node pair set. We denote by $N_{SI}(i)$ the neighboring set for a node i derived from the

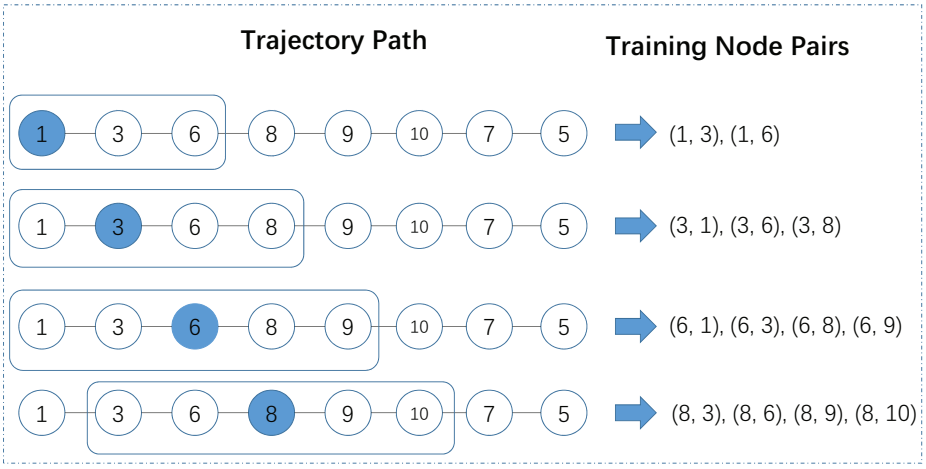


Figure 6.1: Generating node pairs from a trajectory path 1,3,6,8,9,10,7,5. The window size $\omega = 2$ and only the first four nodes 1, 3, 6 and 8 as the center node are illustrated as examples.

SI spreading process. A neighboring node j of i may appear multiple times in $N_{SI}(i)$ if (i, j) appears multiple times in the node pair set.

Let $p(j|i)$ be the probability of observing neighboring node j given node i. We model the conditional probability $p(j|i)$ as the softmax unit parametrized by the product of the embedding vectors, i.e., $\overrightarrow{u_i}$ and $\overrightarrow{u_j}$, as follows:

$$p(j|i) = \log \frac{\exp(\overrightarrow{u_i} \cdot \overrightarrow{u_j}^T)}{\sum_{k \in \mathcal{N}} \exp(\overrightarrow{u_i} \cdot \overrightarrow{u_k}^T)} \tag{6.1}$$

Skip-Gram is to derive the set of the N embedding vectors that maximizes the log probability of observing every neighboring node from $N_{SI}(i)$ for each i. Therefore, one maximizes

$$\max \quad \mathcal{O} = \sum_{i \in \mathcal{N}} \sum_{j \in N_{SI}(i)} \log p(j|i). \tag{6.2}$$

Equation (6.2) can be further simplified to

$$\max \quad \mathcal{O} = \sum_{i \in \mathcal{N}} \left(-\log Z_i + \sum_{j \in N_{SI}(i)} \overrightarrow{u_i} \cdot \overrightarrow{u_k}^T \right), \tag{6.3}$$

where

$$Z_i = \sum_{k \in \mathcal{N}} \exp(\overrightarrow{u_i} \cdot \overrightarrow{u_k}^T). \tag{6.4}$$

To compute Z_i for a given i, we need to traverse the entire node set $\mathcal{N}$, which is computationally costly. To solve this problem, we introduce negative sampling [22], which randomly selects a certain number of nodes k from $\mathcal{N}$ to approximate Z_i. To get the embedding vectors for each node, we use the stochastic gradient ascent to optimize Eq. (6.3).

The static network embedding algorithm proposed above from the SI-spreading-based static network sampling and Skip-Gram model is named as *SINE*.

6.2.3. SI-SPREADING-BASED TEMPORAL NETWORK SAMPLING

We generalize *SINE* to the SI-spreading-based temporal network embedding by deploying SI spreading processes on the given temporal network, namely, *TSINE*. For a temporal network $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, SI spreading follows the time step of the contacts in $\mathcal{G}$. Initially, node i is chosen as the seed of the spreading process. At every time step $t \in [0, T]$, an infected node infects each of its susceptible neighbor in the snapshot through the contact between them with probability β. The process stops at time T. We construct the spreading trajectory starting from node i as $\mathcal{T}_i(\beta)$, which records the union of nodes that get infected together with the contacts through which these nodes get infected. We propose two protocols to select the seed node of the SI spreading. In the first protocol, we start by selecting uniformly at random a node i as the seed. Then, we select uniformly at random a time step from all the times of contacts made by node i as the starting point of the spreading process, i.e., the time when i gets initially infected. We refer to this protocol as *TSINE1*. In the second protocol, we choose a node i uniformly at random as the seed and start the spreading at the time when node i has the first contact. We refer to this protocol as *TSINE2*.

Both *TSINE1* and *TSINE2* generate the node pair set from the spreading trajectory $\mathcal{T}_i(\beta)$ in the same way as described in Section 6.2.1. The node pairs from the node pair set is the input of of Skip-Gram for calculating the embedding vector for each node. The SI-spreading-based temporal network embedding uses the information on the time stamps of contacts in addition to the information used by the static network embedding.

6.3. RESULTS

For the link prediction task in a static network, we remove a certain fraction of links from the given network and predict these missing links based on the remaining links. We apply our static network embedding algorithm to the remaining static network to derive the embedding vectors for the nodes, which are used for link prediction. For a temporal network, we select a fraction of node pairs that have at least one contact. We remove all the contacts between the selected node pairs from the given temporal network. Then, we attempt to predict whether the selected node pairs have at least one contact or not based on the remaining temporal network. We use the area under the curve (AUC) score to evaluate the performance of the algorithms on the link prediction task. The AUC quantifies the probability of ranking a random node pair that is connected or has at least a contact higher than a random node pair that is not connected or has no contact.

6.3.1. EMPIRICAL NETWORKS

We consider temporal networks, each of which records the contacts and their corresponding time stamps between every node pair. For each temporal network $\mathcal{G}$, one can obtain the corresponding static network G by aggregating the contacts between each node pair over time. In other words, two nodes are connected in static network G if there is at least one contact between them in $\mathcal{G}$. The static network G derived from $\mathcal{G}$ is unweighted by definition. We consider the following temporal social network data sets.

- *HT2009* [30] is a network of face-to-face contacts between the attendees of the ACM Hypertext 2009 conference.
- *Manufacturing Email (ME)* [31] is an email contact network between employees in a mid-sized manufacturing company.

- *Haggle* [32] records the physical contacts between individuals via wireless devices.
- *Fb-forum* [33] captures the contacts between students at University of Califonia, Irvine, in a Facebook-like online forum.
- *DNC* [34] is an email contact network in the 2016 Democratic National Committee email leak.
- *CollegeMsg* [35] records messages between the users of an online community of students from the University of California, Irvine.

Table 6.1 provides some properties of the empirical temporal networks. In the first three columns we show the properties of the temporal networks, i.e., the number of nodes (N), timestamps (T) and contacts ($|\mathscr{L}|$). In the remaining columns, we show the properties of the corresponding aggregate static networks, including the number of links ($|E|$), link density, average degree, and clustering coefficient. The temporal networks are considerably different in size, which ranges from hundreds to thousands of nodes, as well as in the network density and clustering coefficient. Choosing networks with different properties allows us to investigate whether the performance of our algorithms can be consistent across networks.

Table 6.1: Properties of the empirical temporal networks. The number of nodes (N), timestamps (T), and contacts ($|\mathscr{L}|$) are shown. In addition, the number of links ($|E|$), link density, average degree, and clustering coefficient of the corresponding static network are shown.

Dataset	N	T	$\lvert\mathscr{L}\rvert$	$\lvert E\rvert$	Link Density	Average Degree	Clustering Coefficient
HT2009	113	5,246	20,818	2,196	0.35	38.87	0.53
ME	167	57,842	82,927	3,251	0.23	38.93	0.59
Haggle	274	15,662	28,244	2,124	0.57	15.5	0.63
Fb-forum	899	33,515	33,720	7,046	0.02	15.68	0.06
DNC	1,891	19,383	39,264	4,465	0.002	4.72	0.21
CollegeMsg	1,899	58,911	59,835	13,838	0.008	14.57	0.11

6.3.2. BASELINE ALGORITHMS

We consider three state-of-the-art network embedding algorithms based on Skip-Gram. These baseline algorithms and the algorithms that we proposed differ only in the method to sample trajectory paths, from which the node pair set, i.e., the input to the Skip-Gram, is derived. *DeepWalk* [20] and *Node2Vec* [23] are static network embedding algorithms based on random walks. *CTDNE* [26] is a temporal network embedding algorithm based on random walks.

- *DeepWalk* [20] deploys classic random walks on a given static network.
- *Node2vec* [23] deploys biased random walks on a given static network. The biased random walk gives a trade-off between breadth-first-like sampling and depth-first-like sampling of the neighborhood, which is controlled via two hyper-parameters p and q. We use a grid search over $p, q \in \{0.01, 0.25, 0.5, 1, 2, 4\}$ to obtain embeddings that achieve the largest AUC value for link prediction.
- *CTDNE* [26]: *CTDNE* is a temporal network embedding algorithm based on temporal random walks. The main idea is that the timestamp of the next temporal contact on

the walk should be larger than the timestamps of previously traversed contacts. Given a temporal network $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, the starting contact for the temporal random walk is selected uniformly at random. Thus, every contact has probability $1/|\mathcal{L}|$ to be selected as the starting contact. Assume that a random walker visits node i at time step t. We define $\Gamma_t(i)$ as the set of nodes that have contacted node i after time t allowing duplicated elements. A node may appear multiple times in $\Gamma_t(i)$ because it may have multiple contacts with node i over the course of time. The next node to walk to is uniformly selected from $\Gamma_t(i)$, i.e., every node in $\Gamma_t(i)$ is chosen with probability $1/|\Gamma_t(i)|$. Nguyen et al. [26] generalized the starting contact and the successor node of a temporal walk to other distributions beyond the uniform distribution illustrated here. When we compare the performance of the algorithms on link prediction, we explore the embeddings that give the largest AUC value for link prediction of *CTDNE* by taking into account all possible generalizations proposed by Nguyen et al.

In our SI-spreading-based algorithms for both static and temporal networks, we set $\beta \in \{0.001, 0.01, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$. We use $\omega = 10$ and embedding dimension $d = 128$ for our algorithms and the baseline algorithms.

6.3.3. PERFORMANCE EVALUATION

TRAINING AND TEST SETS

In this section, we illustrate how to generate the training and test sets in the link prediction task in temporal and static networks. We run the network embedding algorithms on the corresponding training set and obtain embedding vector for each node, and use the AUC to evaluate the link prediction performance in the test set.

Given a temporal network $\mathcal{G}$, we select uniformly at random 75% node pairs among the node pairs that have at least one contact between them in $\mathcal{G}$ as the training set for temporal embedding algorithms, including all the contacts and their timestamps. The training set for static network embedding algorithms is the aggregation of the training set for temporal embedding algorithms. In other words, for every node pair, there is a link between the two nodes in the training set for static network embedding if and only if they have at least one contact in the training set for temporal embedding algorithms.

We use the remaining 25% node pairs among the node pairs that have at least one contact of $\mathcal{G}$ as the positive links in the test set. We label these node pairs 1. Then, we uniformly randomly sample an equal number of node pairs in $\mathcal{G}$ which have no contact between them. These node pairs are used as negative links in the test set, which we label 0. The same test set is used for the link prediction task in both temporal and static networks.

For each temporal network data set, we randomly split the network to obtain the training and test set according to the procedures given above five times. Both random walks and SI spreading processes are stochastic. For each split data, we run each algorithm on the training set and perform the link prediction on the test set for ten realizations. Therefore, we obtain ten AUC scores for each splitting of the data into the training and test sets, evening the randomness stemming from stochasticity of the random walk or SI spreading processes. We obtain the AUC score for each algorithm with a given parameter set as an average over 50 realizations in total.

EVALUATION RESULTS

We summarize the overall performance of the algorithms on missing link prediction in Table 6.2. For each algorithm, we tune the parameters and show the optimal average AUC score.

Table 6.2: AUC scores for link prediction. All the results shown are the average over 50 realizations. Bold indicates the optimal AUC among the embedding algorithms, * indicates the optimal AUC among all the algorithms. L2, L3, L4 are the short for link prediction metrics which counts the number of $l = 2, 3, 4$ paths, respectively.

Dataset	DeepWalk	Node2Vec	CTDNE	TSINE1	TSINE2	SINE	L2	L3	L4
HT2009	0.5209	0.5572	0.6038	0.6740	**0.6819**	0.6726	0.7069*	0.7066	0.7055
ME	0.6439	0.6619	0.6575	0.7329	0.7462	**0.7744**	0.7855	0.7878*	0.7790
Haggle	0.3823	0.7807	0.7796	0.8051	0.8151	**0.8267***	0.8167	0.8255	0.8226
Fb-forum	0.5392	0.6882	0.6942	0.7104	0.7195	**0.7302***	0.5606	0.7179	0.7203
DNC	0.5822	0.5933	0.7274	0.7539	0.7529	**0.7642**	0.7704*	0.7627	0.7193
CollegeMsg	0.5356	0.5454	0.7872	0.8257	0.8321	**0.8368**	0.7176	0.8609*	0.8203

Among the static network embedding algorithms, *SINE* significantly outperforms *DeepWalk* and *Node2Vec*. The improvement in the AUC score is up to 30% on the *CollegeMsg* dataset. Embedding algorithms *CTDNE*, *TSINE1* and *TSINE2* are for temporal networks. The SI-spreading-based algorithms (i.e., *TSINE1* and *TSINE2*) also show better performance than random-walk-based one (*CTDNE*). Additionally, *TSINE2* is slightly better than *TSINE1* on all data sets. Therefore, we will focus on *TSINE2* in the following analysis. In fact, *SINE* shows better performance than temporal network embedding methods including *TSINE2* on all data sets except for *HT2009*. It has been shown that temporal information is important for learning embeddings [26, 36, 37]. However, up to our numerical efforts, *SINE* outperforms the temporal network algorithms although SINE deliberately neglects temporal information.

To get insights into the different performance among the embedding algorithms, we further investigate the distribution of the dot product of node embedding vectors. Given a link (i, j) in the test set, we compute the dot product of the two end nodes' embedding vectors, i.e., $\overrightarrow{u_i} \cdot \overrightarrow{u_j}^T$. We show the dot product distribution for the positive links and negative links in the test set separately. For each embedding algorithm, we consider only the parameter set that maximizes the AUC, i.e., the parameter values with which the results are shown in Table 6.2. We show the distribution of the dot product for *Haggle* in Figure 6.2 and for the other data sets in Figure S1–S5 in the Appendix. Compared to the random-walk-based algorithms, *TSINE2* and *SINE* yield more distinguishable distributions between the positive (grey) and the negative links (pink). This result supports the better performance of SI-spreading-based embeddings than random-walk-based ones.

The embedding algorithms differ only in the sampling method to generate the node pair set. These algorithms use the same Skip-Gram architecture, which takes the node pair set as input, to deduce the embedding vector for each node. We explore further how the algorithms differ in the node pair sets that they sampled. The objective is to discover the relation between the properties of the sampled node pairs and the performance of an embedding method. We represent the node pair set generated by an embedding method as a network $G_S = (\mathcal{N}, E_S)$, so called the sampled network. Two nodes are connected in G_S if they form a node pair in the node pair set. It should be noted that G_S is an unweighted network. For each algorithm, with the parameter set that maximizes the AUC, we show the cumulative degree distribution of its sampled network G_S in Figure 6.3. The cumulative degree distribution of the training set for static network is also given. Compared to the cumulative degree distribution of the training set, the sampled networks tend to have a higher node degree. Zhang et al. and Gao et al. [25, 38] have shown that when the degree distribution of G_S is closer to that of the

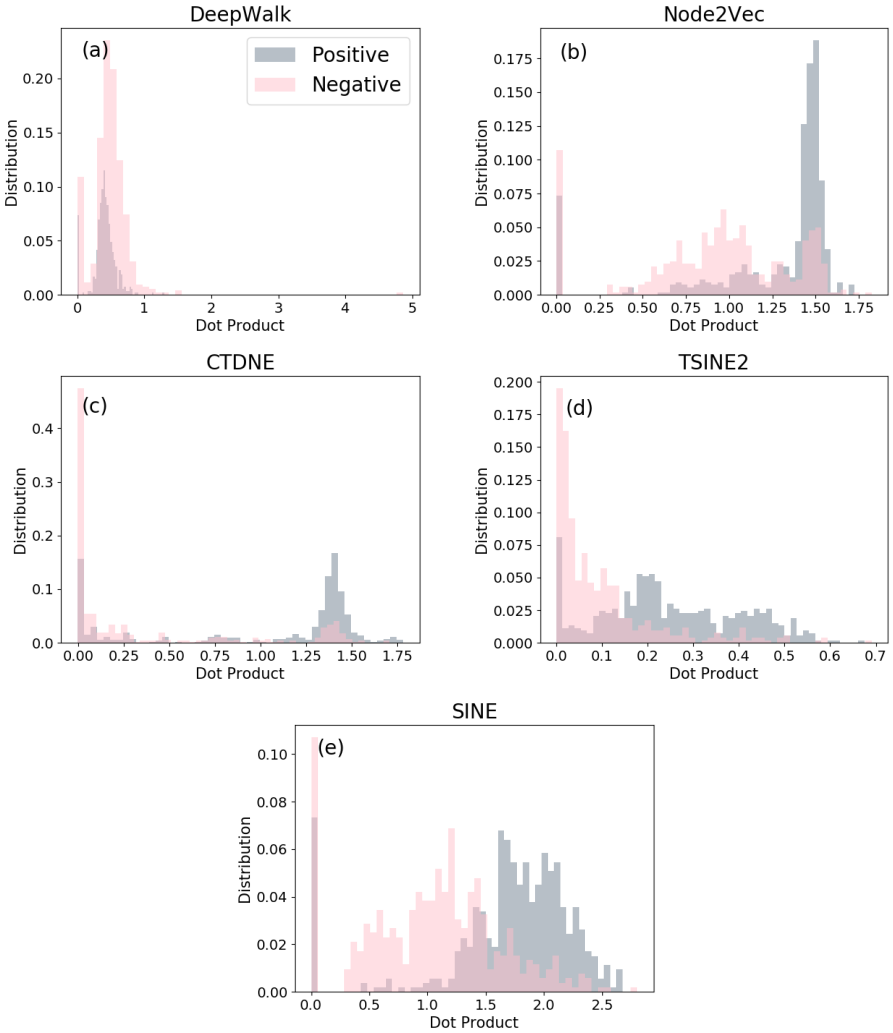


Figure 6.2: The dot product distribution of the two end nodes' embedding vectors of the positive and negative links in the test set. We show the result of the *Haggle* data set. For each algorithm, we use the same parameter settings as that of Table 6.2 to obtain the embeddings. Dot products of positive links are shown in grey. Negative links are shown in pink. The results are shown for algorithms (a) *DeepWalk*; (b) *Node2Vec*; (c) *CTDNE*; (d) *TSINE2* and (e) *SINE*.

training set, the prediction performance of a random-walk-based algorithm tends to be better. Even though SI-spreading based algorithms perform the best across the data sets, we have not found a direct relation between the performance of the embedding algorithm and similarity between the degree distribution of the sampled network and that of the training set.

Similarity-based methods such as the number of $l = 2,3,4$ paths have been used for link prediction problem [9]. A l path between two nodes refers to a path that contains l links. We show examples of $l = 2,3,4$ path between a node pair i and j in Figure 6.4. Kovács et al. [39] have shown that l paths ($l = 3,4$) outperform existing link prediction methods in

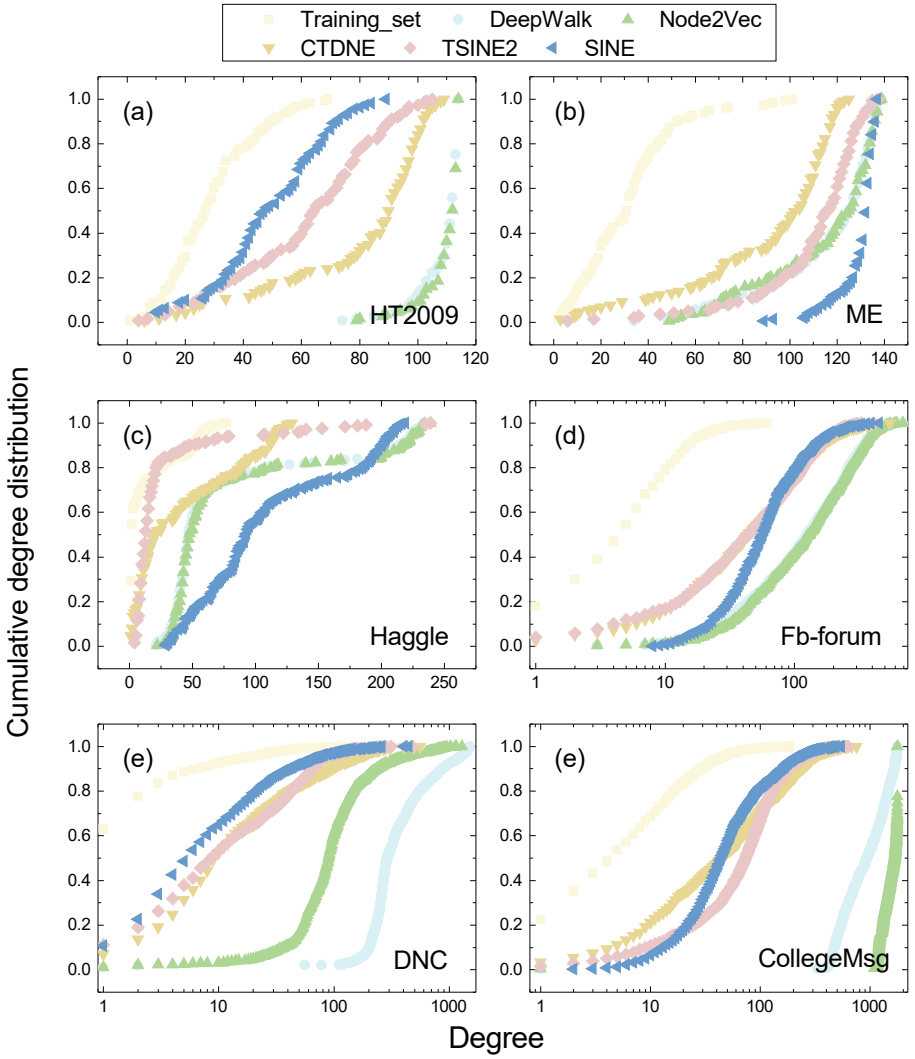


Figure 6.3: Cumulative degree distribution of the static network derived from the training set and that of the sampled networks G_S from different algorithms. We show the results for (a)*HT2009*; (b)*ME*; (c)*Haggle*; (d)*Fb-forum*; (e)*DNC*; (f)*CollegeMsg*.

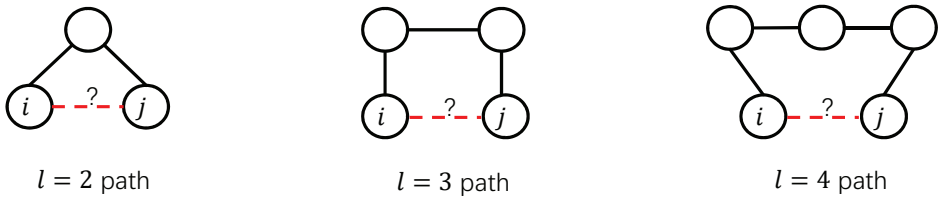


Figure 6.4: Illustration of l paths between a pair of nodes i and j. Here we show $l = 2, 3, 4$.

predicting protein interaction. Cao et al. [40] found that network embedding algorithms based on random walks sometimes perform worse in link prediction than the number of $l = 2$ paths or equivalently the number of common neighbors. This result suggests a limit of random-walk-based embedding in identifying the links between node pairs that have many common neighbors. Therefore, we explore further whether our SI-spreading-based algorithms can overcome this limitation, thus possibly explain their outperformance.

We investigate what kind of network structure surrounding links makes them more easily be predicted. For every positive link in the test set, we study its two end nodes' topological properties (i.e., the number of $l = 2$, $l = 3$ and $l = 4$ paths) and the dot product of the embedding vectors of its two end nodes. Given a network, the parameters of each embedding algorithm are tuned to maximize the AUC, as given in Table 2. We take the data set *Haggle* as an example. Figure 6.5 show the relation between the dot product of the embedding vectors and the number of $l = 2, 3, 4$ paths of the two end nodes of a positive link in the test set for all the embedding methods. The Pearson correlation coefficient (PCC) between the two variables for all the networks and algorithms is given in Table S1 in the Appendix. Figure 6.5 and Table S1 together show that the dot product of the embedding vectors constructed from *TSINE2* and *SINE* is more strongly correlated with the number of l paths, where $l = 2$, 3 or 4, than the random-walk-based embeddings. This result suggests that SI-spreading-based algorithms may better predict the links whose two end nodes have many l-paths, thus overcoming the limit of random-walk-based embedding algorithms.

The number of $l = 2, 3$ paths has been used to predict links in [9, 39, 40]. The observation and the limit of random-walk-based embedding algorithms motivate us to use the number of $l = 2, 3, 4$ paths between a node pair to predict the missing links. Take $l = 2$ paths as an example. For every link in the test set, the number of $l = 2$ paths between the two end nodes in the training set is used to estimate the likelihood of connection between them. In the networks we considered, two end nodes of a link tend to be connected by $l = 2$, $l = 3$ and $l = 4$ paths (see Figures 6.5). Table 6.2 ($L2, L3, L4$ shown in the table correspond to the method of using the number of $l = 2, 3, 4$ path for link prediction) shows that in such networks, the similarity-based methods do not evidently outperform the SI-spreading-based embedding. Actually, the SI-spreading-based embedding performs better in two out of six networks.

Next, we study the effect of the sampling size, B, on the performance of each algorithm. The sampling size is quantified as the the total length of the trajectory paths as defined in Section 6.2.1. Given a network, we set $\mathscr{B} = NX$, where N is the size of the network and $X \in \{1, 2, 5, 10, 25, 50, 100, 150\}$. We evaluate our SI-spreading-based embedding algorithms *SINE* and *TSINE2*, and one random-walk-based embedding algorithm *CTDNE*, because *CTDNE* performs mostly the best among all random-walk-based algorithms. The result is shown in Figure 6.6. For each X, we tune the other parameters to show the optimal AUC in the figure. Both *SINE* and *TSINE2* perform better than *CTDNE* and are relatively insensitive to the sampling size. This means that they achieve a good performance even when the sampling size is small, even with $X = 1$. The random-walk-based algorithm, *CTDNE*, however, requires a relatively large sampling size to achieve a comparable performance with *SINE* and *TSINE2*.

Finally, the AUC as a function of the infection probability, β, is shown in Figure 6.7. For each β, we tune the other parameters to show the optimal AUC. The SI-spreading-based algorithms achieve high performance with a small infection probability ($0.001 \leq \beta \leq 0.1$) for all the data sets. The high performance of SI-spreading-based embedding algorithms with the small value of X and β across different networks motivates the further study whether one can optimize the performance by searching a smaller range of the parameter values.

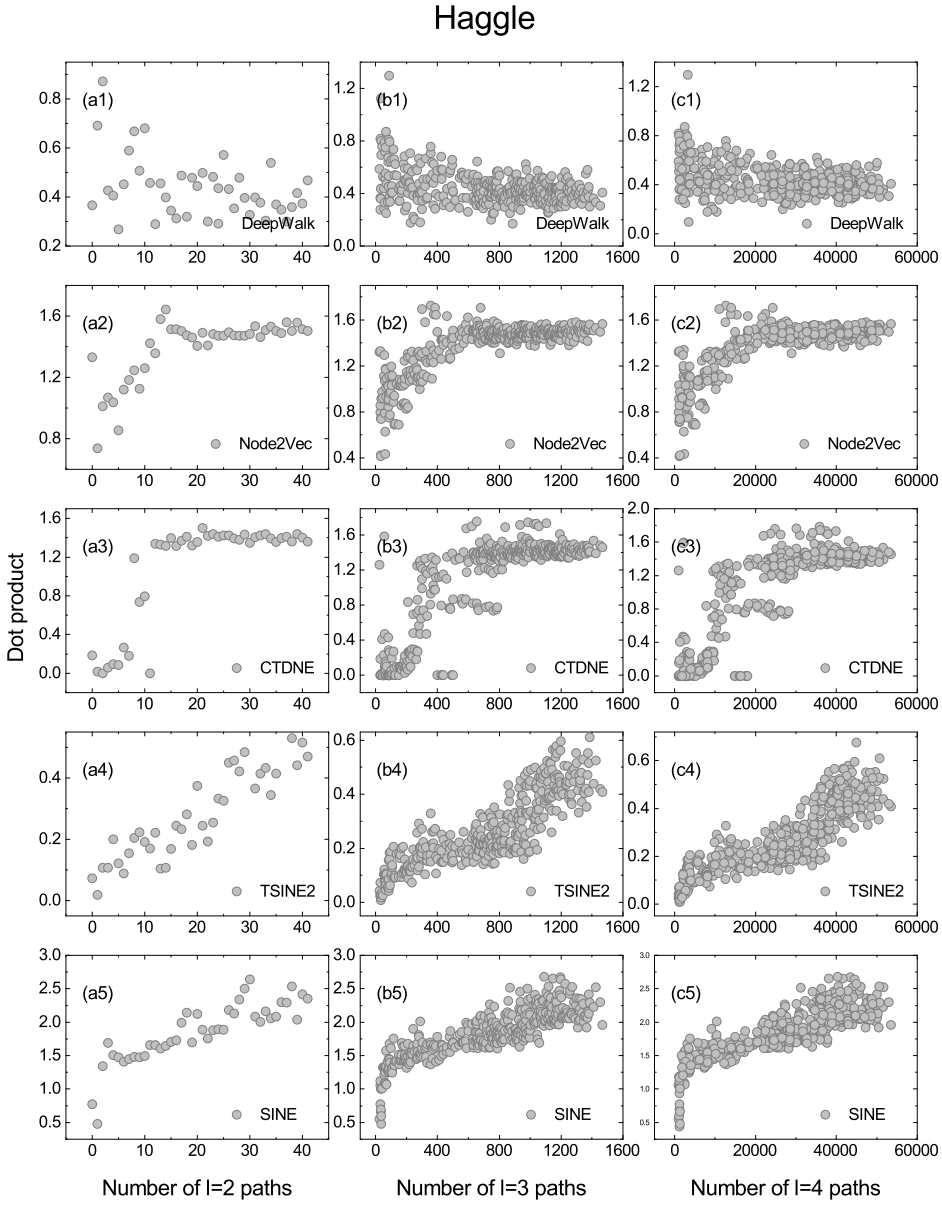


Figure 6.5: Relation between the dot product of the two end nodes' embedding vectors and the number of $l = 2, 3, 4$ paths between the two end nodes of the positive links in the test set for *Haggle* data set. (a1–a5), (b1–b5) and (c1–c5) are the results for the number of $l = 2, 3, 4$ paths, respectively.

6.4. CONCLUSIONS

In this chapter, we proposed network embedding algorithms based on SI spreading processes in contrast to the previously proposed embedding algorithms based on random walks [41, 42]. We further evaluated the embedding algorithms on the missing link prediction task. The key

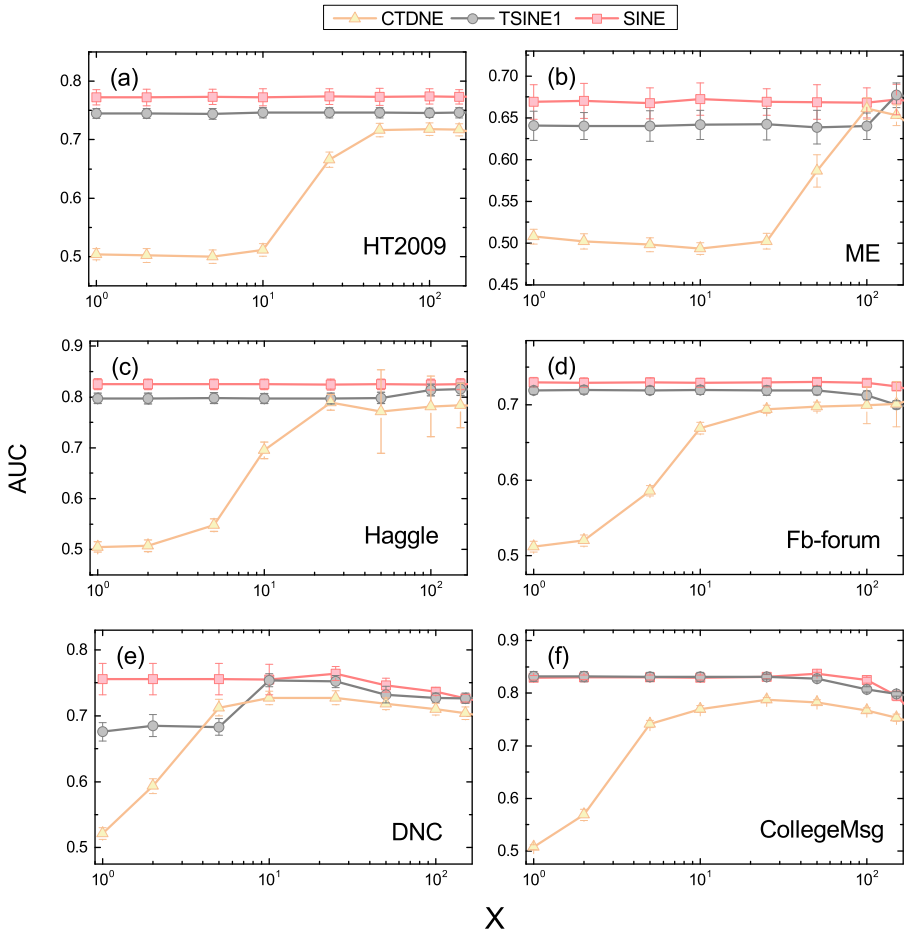


Figure 6.6: Influence of the sampling size $B = NX$ on the link prediction performance, i.e., AUC score. The error bar shows the standard deviation of the AUC score calculated on the basis of 50 realizations. We show the results for (a)*HT2009*; (b)*ME*; (c)*Haggle*; (d)*Fb-forum*; (e)*DNC*; (f)*CollegeMsg*.

point of an embedding algorithm is how to design a strategy to sample trajectories to obtain embedding vectors for nodes. We used the SI model to this end. The algorithms that we proposed are *SINE* and *TSINE*, which use static and temporal networks, respectively.

On six empirical data sets, the SI-spreading-based network embedding algorithm on the static network, i.e., *SINE*, gains much more improvement than state-of-the-art random-walk-based network embedding algorithms across all the data sets. The SI-spreading-based network embedding algorithms on the temporal network, *TSINE1* and *TSINE2*, also show better performance than the temporal random-walk-based algorithm. Temporal information provides additional information that may be useful for constructing embedding vectors [26, 36, 37]. However, we find that *SINE* outperforms *TSINE*, which uses timestamps of the contacts. This result suggests that temporal information does not necessarily improve the embedding for missing link prediction. Moreover, when the sampling size of the Skip-Gram is small, the performance of the SI-spreading-based embedding algorithms is still high.

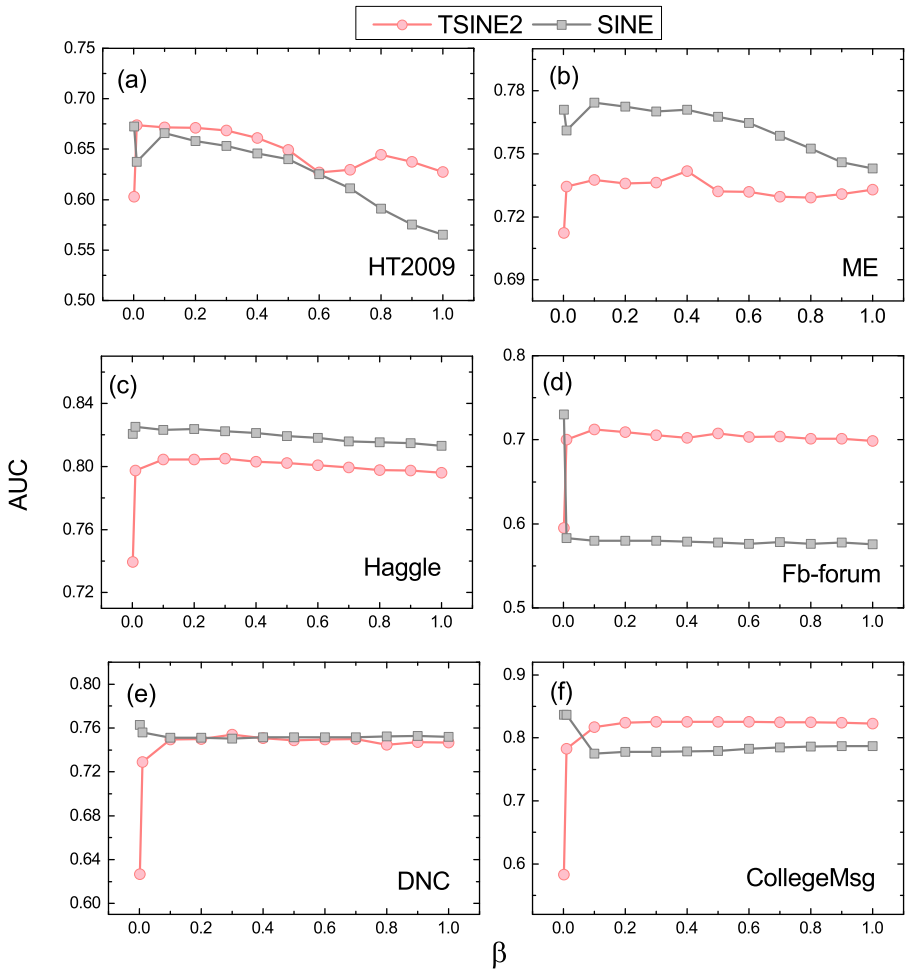


Figure 6.7: AUC as a function of β. We show the results for (a)*HT2009*; (b)*ME*; (c)*Haggle*; (d)*Fb-forum*; (e)*DNC*; (f)*CollegeMsg*.

Sampling trajectory paths takes time especially for large-scale networks. Therefore, our observation that the SI-spreading-based algorithms require less samples than other algorithms promises the applicability of the SI-spreading-based algorithms to larger networks than the random-walk-based algorithms. Finally, we show insights of why SI-spreading-based embedding algorithms performs the best by investigating what kind of links are likely to be predicted.

We deem that the following future work as important. We have already applied susceptible-infected-susceptible (SIS) model and evaluated the SIS-spreading-based embedding. However, this generalization has not improved the performance in the link prediction task. Therefore, one may explore whether or not sampling the network information via the other spreading processes, such as susceptible-infected-recovered (SIR) model, further improves the embedding. It is also interesting to explore further the performance of the SI-spreading-based algorithms in other tasks such as classification and visualization. Moreover, the SI-spreading-based sampling strategies can also be generalized to other types of networks, e.g., directed networks, signed networks, and multilayer networks.

6.5. APPENDIX

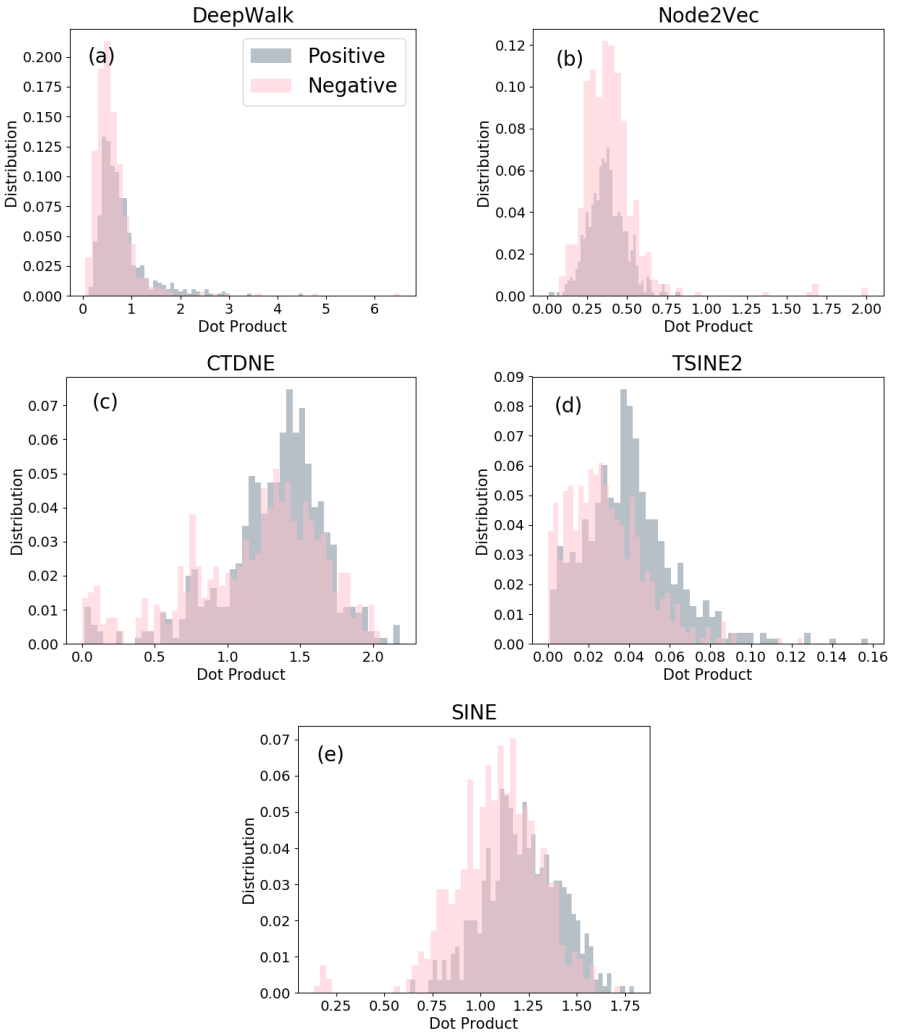


Figure 6.8: The dot product distribution of the end nodes' embedding vectors of the positive and negative links in the test set. We show the result of the *HT2009* dataset. For each algorithm, we use the same parameter settings as that of Table 2 in the main manuscript to get the embeddings. Dot products of positive links are in grey. Negative links are in pink.

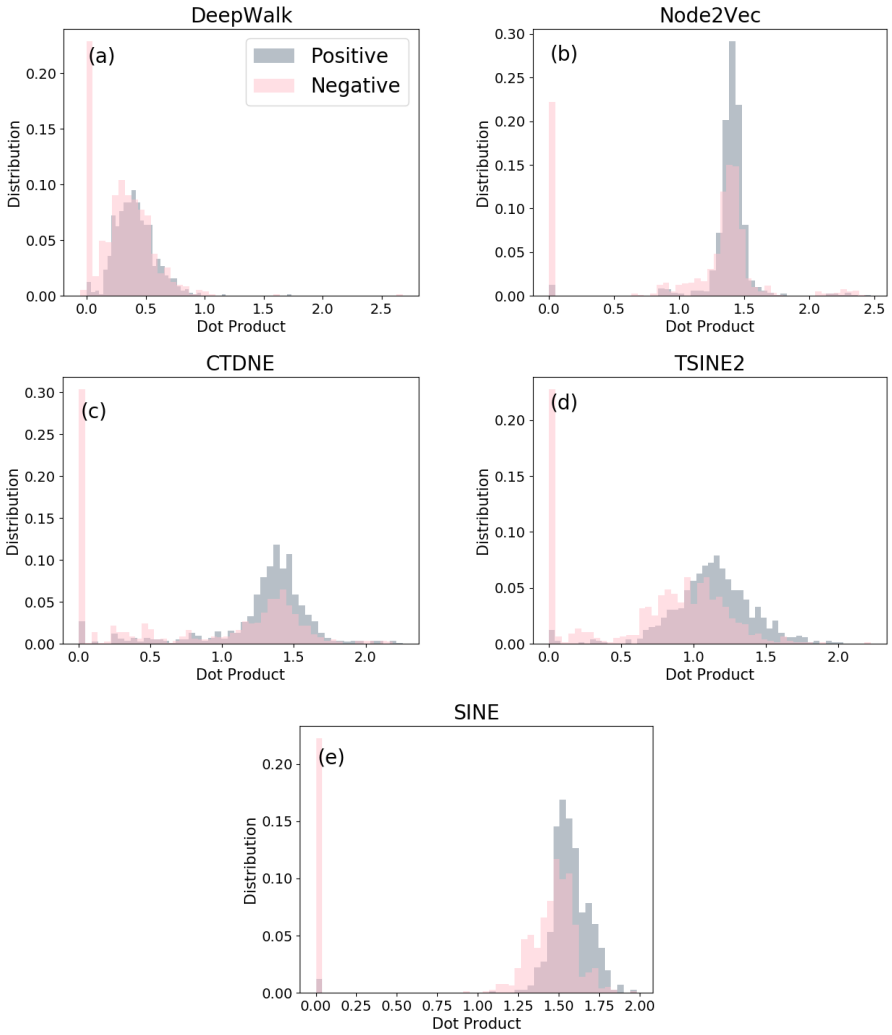


Figure 6.9: Distribution of the dot product of the end nodes' embedding vectors of the positive and negative links in the test set. We show the result of the *ME* data set. For each embedding algorithm, we use the same parameter values as those used for Table 2 in the main manuscript. The distribution for the positive and negative links are shown in grey and pink, respectively.

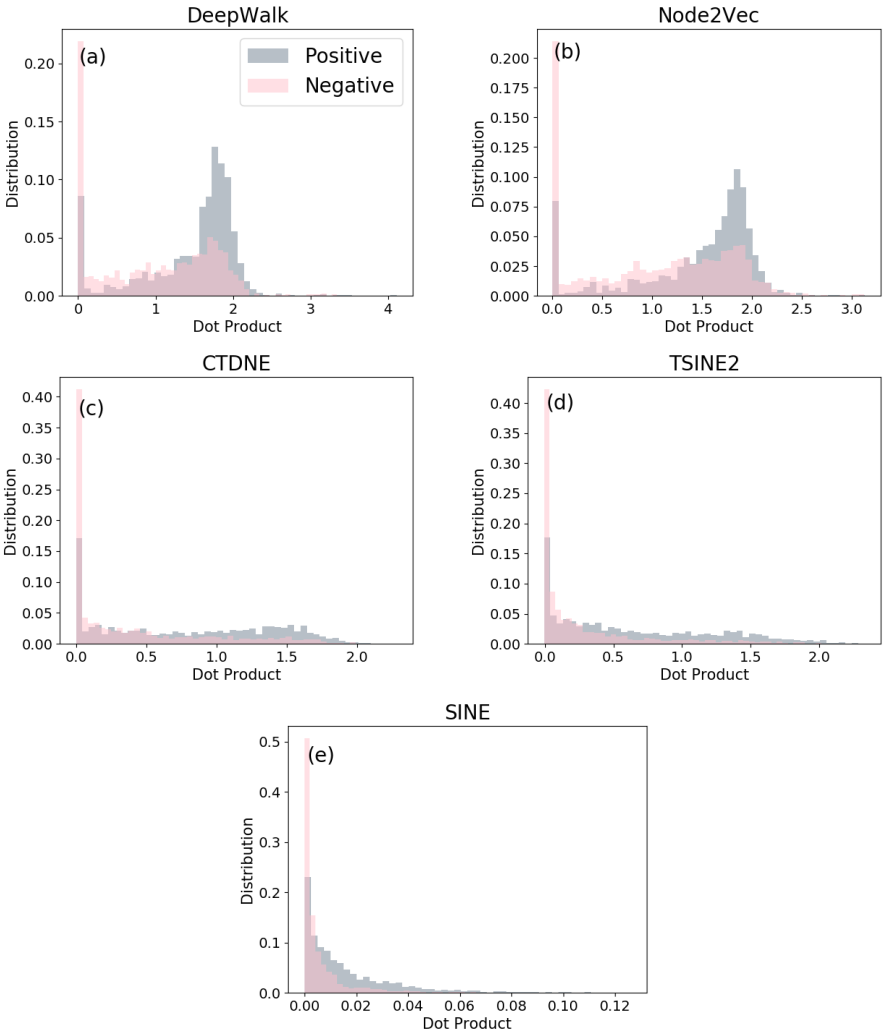


Figure 6.10: Distribution of the dot product of the end nodes' embedding vectors of the positive and negative links in the test set. We show the result of the *Fb-forum* data set. For each embedding algorithm, we use the same parameter values as those used for Table 2 in the main manuscript. The distribution for the positive and negative links are shown in grey and pink, respectively.

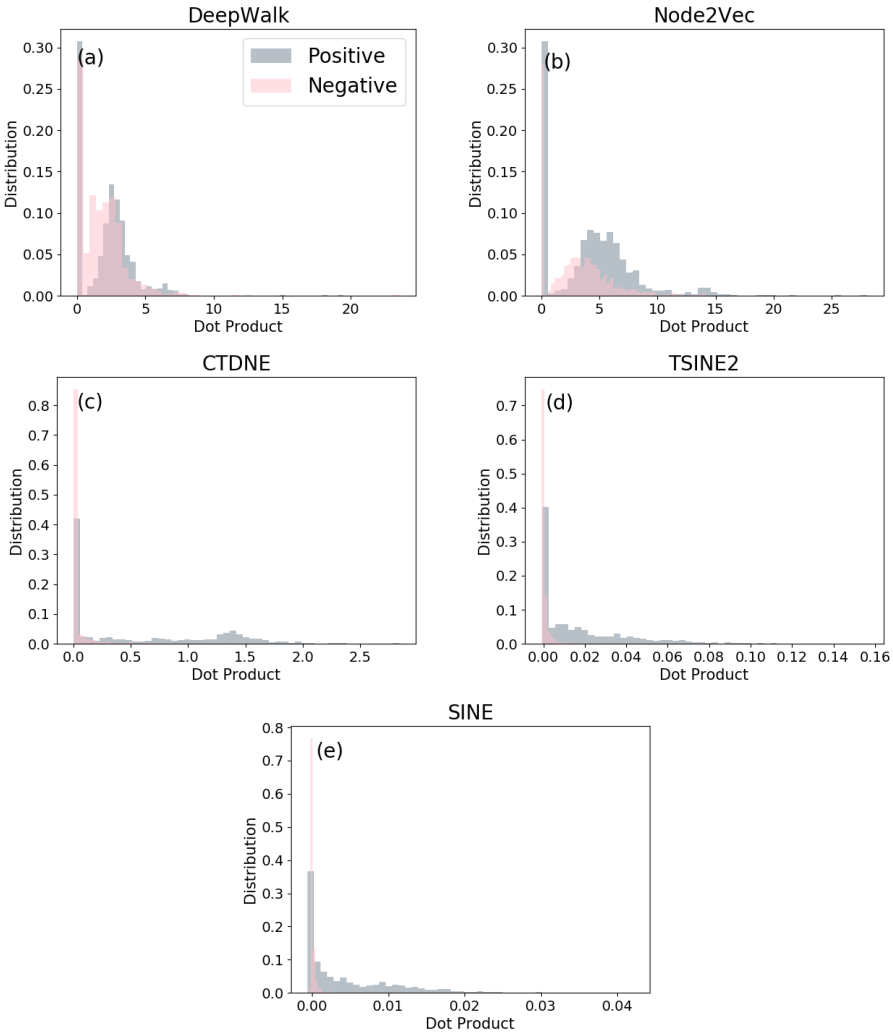


Figure 6.11: Distribution of the dot product of the end nodes' embedding vectors of the positive and negative links in the test set. We show the result of the *DNC* data set. For each embedding algorithm, we use the same parameter values as those used for Table 2 in the main manuscript. The distribution for the positive and negative links are shown in grey and pink, respectively.

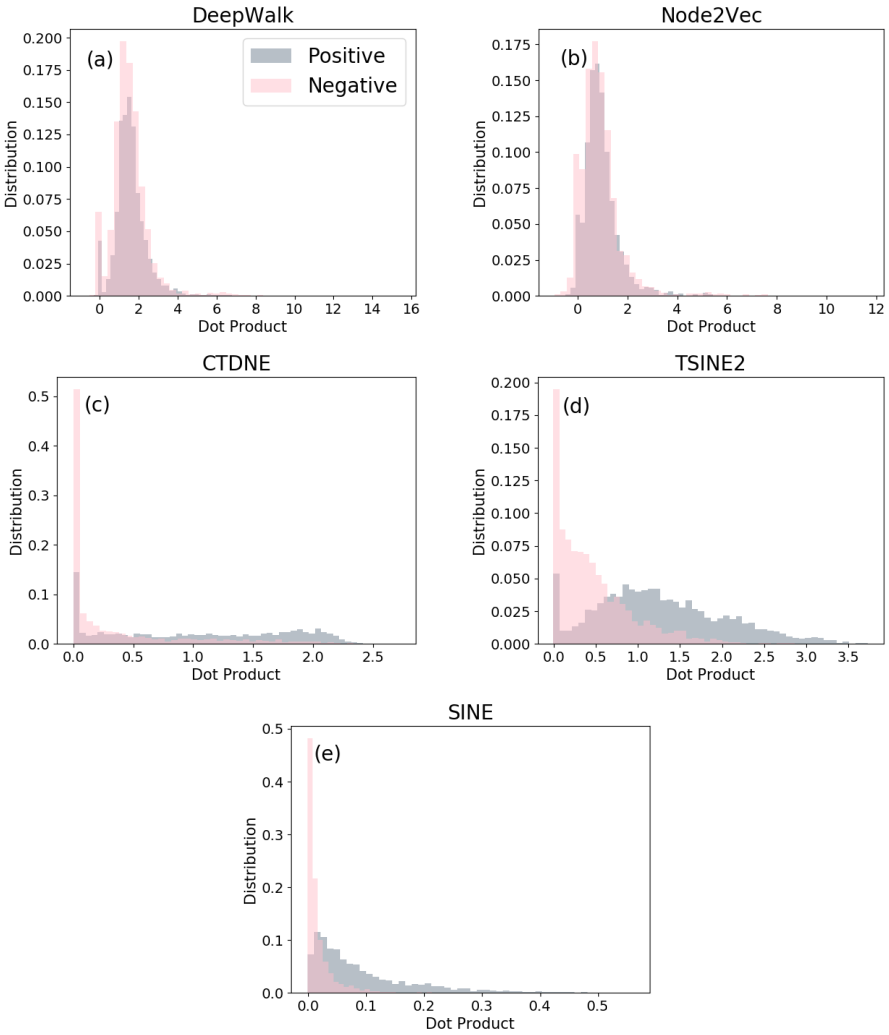


Figure 6.12: Distribution of the dot product of the end nodes' embedding vectors of the positive and negative links in the test set. We show the result of the *CollegeMsg* data set. For each embedding algorithm, we use the same parameter values as those used for Table 2 in the main manuscript. The distribution for the positive and negative links are shown in grey and pink, respectively.

Table 6.3: Pearson correlation coefficient between the number of $l = 2,3,4$ paths and the dot product of node pairs' embedding vector, compared across the different algorithms. The node pairs are the positive links from the test set. We count the number of $l = 2,3,4$ paths of each positive link from the training set. To produce the embedding vectors, we used the same parameter values as those used for Table 2 in the main manuscript.

Metric	Dataset	DeepWalk	Node2vec	CTDNE	TSINE2	SINE
the number of $l = 2$ paths	HT2009	-0.1476 ± 0.0462	0.1466 ± 0.0552	0.3531 ± 0.0484	0.7440 ± 0.0439	0.7824 ± 0.0326
	ME	0.0627 ± 0.0347	0.1465 ± 0.0889	0.4593 ± 0.0556	0.6120 ± 0.0410	0.6324 ± 0.0327
	Haggle	-0.3476 ± 0.0443	0.6863 ± 0.0416	0.8098 ± 0.0122	0.8656 ± 0.0139	0.8564 ± 0.0112
	Fb-forum	0.2099 ± 0.0663	0.1534 ± 0.0272	0.2152 ± 0.0329	0.3322 ± 0.0561	0.4538 ± 0.0586
	DNC	-0.0125 ± 0.0223	0.0263 ± 0.0196	0.5011 ± 0.0836	0.6537 ± 0.0718	0.7220 ± 0.0754
	CollegeMsg	0.0334 ± 0.0159	0.0521 ± 0.0171	0.3674 ± 0.0191	0.5648 ± 0.0133	0.6328 ± 0.0219
the number of $l = 3$ paths	HT2009	-0.2323 ± 0.0401	0.0112 ± 0.0383	0.3746 ± 0.0493	0.7736 ± 0.0495	0.8231 ± 0.0314
	ME	-0.1809 ± 0.0371	0.1430 ± 0.0880	0.4494 ± 0.0600	0.6502 ± 0.0470	0.7051 ± 0.0325
	Haggle	-0.3973 ± 0.0419	0.7165 ± 0.0420	0.8372 ± 0.0125	0.8593 ± 0.0146	0.8635 ± 0.0109
	Fb-forum	-0.0664 ± 0.0541	0.3049 ± 0.0194	0.4452 ± 0.0177	0.6465 ± 0.0383	0.8081 ± 0.0146
	DNC	-0.1351 ± 0.0197	-0.1246 ± 0.0162	0.6010 ± 0.0420	0.7762 ± 0.0249	0.8732 ± 0.0158
	CollegeMsg	-0.2611 ± 0.0073	-0.2292 ± 0.0080	0.5338 ± 0.0161	0.7731 ± 0.0090	0.8546 ± 0.0123
the number of $l = 4$ paths	HT2009	-0.2401 ± 0.0413	0.0027 ± 0.0382	0.3810 ± 0.0495	0.7740 ± 0.0502	0.8255 ± 0.0313
	ME	-0.1894 ± 0.0393	0.1479 ± 0.0907	0.4607 ± 0.0598	0.6529 ± 0.0467	0.6989 ± 0.0316
	Haggle	-0.3977 ± 0.0417	0.7163 ± 0.0418	0.8373 ± 0.0122	0.8629 ± 0.0141	0.8660 ± 0.0109
	Fb-forum	-0.1459 ± 0.0678	0.3131 ± 0.0192	0.4525 ± 0.0207	0.6498 ± 0.0414	0.8557 ± 0.0115
	DNC	-0.1368 ± 0.0178	-0.1288 ± 0.0134	0.5821 ± 0.0432	0.7733 ± 0.0226	0.8690 ± 0.0407
	CollegeMsg	-0.2317 ± 0.0115	-0.2017 ± 0.0124	0.5115 ± 0.0140	0.7608 ± 0.0083	0.8542 ± 0.0168

REFERENCES

[1] M. E. Newman, *The structure and function of complex networks,* SIAM review **45**, 167 (2003).

[2] Z.-K. Zhang, C. Liu, X.-X. Zhan, X. Lu, C.-X. Zhang, and Y.-C. Zhang, *Dynamics of information diffusion and its applications on complex networks,* Physics reports **651**, 1 (2016).

[3] L. d. F. Costa, O. N. Oliveira Jr, G. Travieso, F. A. Rodrigues, P. R. Villas Boas, L. Antiqueira, M. P. Viana, and L. E. Correa Rocha, *Analyzing and modeling real-world phenomena with complex networks: a survey of applications,* Advances in Physics **60**, 329 (2011).

[4] Y. Qi, Z. Bar-Joseph, and J. Klein-Seetharaman, *Evaluation of different biological data and computational classification methods for use in protein interaction prediction,* Proteins: Structure, Function, and Bioinformatics **63**, 490 (2006).

[5] M. Girvan and M. E. Newman, *Community structure in social and biological networks,* Proceedings of the national academy of sciences **99**, 7821 (2002).

[6] Y. Jacob, L. Denoyer, and P. Gallinari, *Learning latent representations of nodes for classifying in heterogeneous social networks,* in *Proceedings of the 7th ACM international conference on Web search and data mining* (ACM, 2014) pp. 373–382.

[7] A. L. Traud, P. J. Mucha, and M. A. Porter, *Social structure of facebook networks,* Physica A: Statistical Mechanics and its Applications **391**, 4165 (2012).

[8] D. Liben-Nowell and J. Kleinberg, *The link-prediction problem for social networks,* Journal of the American society for information science and technology **58**, 1019 (2007).

[9] L. Lü and T. Zhou, *Link prediction in complex networks: A survey,* Physica A: statistical mechanics and its applications **390**, 1150 (2011).

[10] L. Lü, M. Medo, C. H. Yeung, Y.-C. Zhang, Z.-K. Zhang, and T. Zhou, *Recommender systems,* Physics reports **519**, 1 (2012).

[11] V. Martínez, F. Berzal, and J.-C. Cubero, *A survey of link prediction in complex networks,* ACM Computing Surveys (CSUR) **49**, 69 (2017).

[12] C. Liu, Y. Ma, J. Zhao, R. Nussinov, Y.-C. Zhang, F. Cheng, and Z.-K. Zhang, *Computational network biology: Data, model, and applications,* Physics Reports (2019).

[13] L. Getoor and C. P. Diehl, *Link mining: a survey,* Acm Sigkdd Explorations Newsletter **7**, 3 (2005).

[14] P. Cui, X. Wang, J. Pei, and W. Zhu, *A survey on network embedding,* IEEE Transactions on Knowledge and Data Engineering (2018).

[15] X. Wang, P. Cui, J. Wang, J. Pei, W. Zhu, and S. Yang, *Community preserving network embedding,* in *Thirty-First AAAI Conference on Artificial Intelligence* (2017).

[16] J. B. Tenenbaum, V. De Silva, and J. C. Langford, *A global geometric framework for nonlinear dimensionality reduction,* science **290**, 2319 (2000).

[17] S. T. Roweis and L. K. Saul, *Nonlinear dimensionality reduction by locally linear embedding,* science **290**, 2323 (2000).

[18] M. Belkin and P. Niyogi, *Laplacian eigenmaps and spectral techniques for embedding and clustering,* in *Advances in neural information processing systems* (2002) pp. 585–591.

[19] G. H. Golub and C. Reinsch, *Singular value decomposition and least squares solutions,* in *Linear Algebra* (Springer, 1971) pp. 134–151.

[20] B. Perozzi, R. Al-Rfou, and S. Skiena, *Deepwalk: Online learning of social representations,* in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD'14 (ACM, New York, NY, USA, 2014) pp. 701–710.

[21] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, *Line:large-scale information network embedding,* in *Proceedings of the 24th International Conference on World Wide Web*, WWW'15 (International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 2015) pp. 1067–1077.

[22] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, *Distributed representations of words and phrases and their compositionality,* in *Proceedings of the 26th International Conference on Neural Information Processing Systems*, NIPS'13, Vol. 2 (Curran Associates Inc., USA, 2013) pp. 3111–3119.

[23] A. Grover and J. Leskovec, *node2vec: Scalable feature learning for networks,* in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining* (ACM, 2016) pp. 855–864.

[24] Z. Cao, L. Wang, and G. de Melo, *Link prediction via subgraph embedding-based convex matrix completion,* in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI 2018). AAAI Press* (2018).

[25] Y. Zhang, Z. Shi, D. Feng, and X.-X. Zhan, *Degree-biased random walk for large-scale network embedding,* Future Generation Computer Systems **100**, 198 (2019).

[26] G. H. Nguyen, J. B. Lee, R. A. Rossi, N. K. Ahmed, E. Koh, and S. Kim, *Continuous-time dynamic network embeddings,* in *Companion of the The Web Conference 2018 on The Web Conference 2018* (International World Wide Web Conferences Steering Committee, 2018) pp. 969–976.

[27] S. Yuan, X. Wu, and Y. Xiang, *Sne: signed network embedding,* in *Pacific-Asia conference on knowledge discovery and data mining* (Springer, 2017) pp. 183–195.

[28] A. Bagavathi and S. Krishnan, *Multi-net: a scalable multiplex network embedding framework,* in *International Conference on Complex Networks and their Applications* (Springer, 2018) pp. 119–131.

[29] C. Qu, X. Zhan, G. Wang, J. Wu, and Z.-k. Zhang, *Temporal information gathering process for node ranking in time-varying networks,* Chaos: An Interdisciplinary Journal of Nonlinear Science **29**, 033116 (2019).

[30] L. Isella, J. Stehlé, A. Barrat, C. Cattuto, J.-F. Pinton, and W. Van den Broeck, *What's in a crowd? analysis of face-to-face behavioral networks,* Journal of theoretical biology **271**, 166 (2011).

[31] R. Michalski, S. Palus, and P. Kazienko, *Matching organizational structure and social network extracted from email communication,* in *International Conference on Business Information Systems* (Springer, 2011) pp. 197–206.

[32] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, *Impact of human mobility on opportunistic forwarding algorithms,* IEEE Transactions on Mobile Computing , 606 (2007).

[33] T. Opsahl, *Triadic closure in two-mode networks: Redefining the global and local clustering coefficients,* Social Networks **35**, 159 (2013).

[34] *Dnc emails network dataset – KONECT,* (2017).

[35] T. Opsahl and P. Panzarasa, *Clustering in weighted networks,* Social networks **31**, 155 (2009).

[36] Y. Zuo, G. Liu, H. Lin, J. Guo, X. Hu, and J. Wu, *Embedding temporal network via neighborhood formation,* in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (ACM, 2018) pp. 2857–2866.

[37] L. Zhou, Y. Yang, X. Ren, F. Wu, and Y. Zhuang, *Dynamic network embedding by modeling triadic closure process,* in *Thirty-Second AAAI Conference on Artificial Intelligence* (2018).

[38] M. Gao, L. Chen, X. He, and A. Zhou, *Bine: Bipartite network embedding,* in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval* (2018) pp. 715–724.

[39] I. A. Kovács, K. Luck, K. Spirohn, Y. Wang, C. Pollis, S. Schlabach, W. Bian, D.-K. Kim, N. Kishore, T. Hao, *et al.*, *Network-based prediction of protein interactions,* Nature communications **10**, 1 (2019).

[40] R.-M. Cao, S.-Y. Liu, and X.-K. Xu, *Network embedding for link prediction: The pitfall and improvement,* Chaos: An Interdisciplinary Journal of Nonlinear Science **29**, 103102 (2019).

[41] X.-X. Zhan, A. Hanjalic, and H. Wang, *Information diffusion backbones in temporal networks,* Scientific reports **9**, 6798 (2019).

[42] X.-X. Zhan, C. Liu, G. Zhou, Z.-K. Zhang, G.-Q. Sun, J. J. Zhu, and Z. Jin, *Coupling dynamics of epidemic spreading and information diffusion on complex networks,* Applied Mathematics and Computation **332**, 437 (2018).

7

REFLECTIONS AND RECOMMENDATIONS

In this thesis, we focus on the impact of temporal network structure on information diffusion process via the study of local and global surrounding properties of nodes and links. Additionally, we propose network representation learning algorithms to study the similarity between network nodes. We present our methodologies, findings and empirical results in five technical chapters. To close the loop back to the introduction, we answer the research questions and reflect on the limitations and practical implications of our thesis in Section 7.1. Based on the obtained insight and observed limitations of the research reported in the thesis, we point out possible directions for future work in Section 7.2.

7.1. MAIN CONTRIBUTION AND REFLECTIONS

In Chapter 2, we investigate the SI diffusion process on a temporal network, where we propose information diffusion backbone to characterize the probability of a link to appear in a diffusion trajectory. Our objective is to explore which kind of links tend to have a non-zero probability to appear a diffusion trajectory, i.e., appear in the backbone. This is unraveled by the exploration of using a wide range of link properties in identifying the links in the backbone. We propose a time-scaled link weight that describes which links will appear in the diffusion backbone besides considering link centrality metrics from static networks. Compared to the metrics from the aggregated static networks, the time-scaled link weight shows evidently better identification power across different empirical network datasets. This implies the importance of considering the temporal information of a link in predicting its appearance in a diffusion trajectory. Despite that the time-scaled weight shows better identification performance than the static metrics, we only identify the link appearance in the backbone. Identifying the ranking of links in their probability to appear in a diffusion trajectory is beyond the scope of the research reported in this chapter.

To emphasize the importance of link ranking in the diffusion process, we further study in Chapter 3 how removing the links with high ranking could help suppress information diffusion. We propose a comprehensive set of link blocking strategies, which capture diverse link properties, including static ones and temporal ones. The temporal ones are mainly based on the information diffusion backbone, which are proposed in Chapter 2. The static ones mainly contain the static properties of a link, such as degree product, node strength product and static betweenness. In great contrast to the previous work, the links with high ranking values are removed in a certain given time period. This mimics a real-world case, in which societal measures for preventing spreading, such as social distancing, are expected to be temporal rather than permanent. We find that four metrics outperform the random blocking, i.e., static betweenness and three other ones based on information diffusion backbone. However, we find that the effectiveness of the blocking strategies depends on the time scale of the networks. The metrics based on the information diffusion backbone tend to perform well in networks that have fast information propagation speed. Static betweenness is a suitable link blocking strategy for networks with a slow spreading speed. We obtain universal performance of these four link blocking strategies for various empirical network datasets, blocking periods and fractions of links. Our findings suggest that as the time scale of temporal network changes, the way how to design link blocking strategy may also need to be updated.

A node is influential in an information diffusion process if the information that starts spreading from it can be widely spread. We model the information diffusion process on temporal networks by an SIR model in Chapter 4. The spreading capacity of a node is defined as the final spreading size of the spreading process started from the node. In a temporal

network, the spreading process follows the temporal paths. Based on this assumption, we propose a temporal information gathering process (TIG-process) to identify influential nodes in an SIR spreading process. In the TIG-process, we iteratively gather information from different orders of neighbors through the temporal paths. We propose two different temporal paths, i.e., fast arrival path and temporal shortest path. Compared to the benchmark metrics, TIG-process outperforms the rest across different temporal network datasets. We only need a small number of iterations to achieve the best identification performance across different empirical networks. Our work illustrates the potential to more precisely identify influential nodes via encoding the high order neighborhood and temporal information.

The studies in Chapter 2, 3 and 4 unravel the importance of temporal information in determining link (node) importance in information diffusion processes on a temporal network. The success of the methods we propose for identifying important links (nodes) relies on the fact that we systematically consider different properties of the links (nodes). Particularly, we take into account the order of the temporal links as well as the nodes that appear in a temporal network. This may inspire the way to tackle optimization problems, such as which node pairs should be stimulated to link and at what time in order to maximize the prevalence of information or epidemic propagation. In addition to the spreading models, i.e., SI and SIR model, that are used in these three chapters, spreading processes of information or epidemic, can be also modeled by a susceptible-infected-susceptible (SIS) model, independent cascade model [1], the threshold model [2] or opinion model [3]. Due to the different governing mechanisms of different spreading models, the role of a link (node) in these spreading processes may differ significantly.

The methodologies that we have proposed on how to identify links that probably appear in an SI spreading trajectory, how to develop link blocking strategies to suppress the spreading of (mis)information and how to identify the influence of a node as a seed node may shed light on how to design the study of link's (node's) role in other spreading processes.

Furthermore, while we limit ourselves to the study of link's (node's) role in a spreading process on temporal networks, the technique studies are also applicable to a range of different disciplines as well as networks.

The deep study of information diffusion processes on temporal networks navigates us to investigate how to use the spreading processes in the network embedding algorithms in Chapter 5 and 6. In Chapter 5, we design a degree-biased-random-walk network embedding algorithm, i.e., *DiaRW*, in a static network. We use degree centrality to represent node importance and design a random walk that tends to visit the surrounding structure of a high degree node more. In addition, we make the length of a walk positively correlated with the degree of the source node. In other words, if the source node has a higher degree, a walk that starts from it has a longer walk length. We obtain each node's embedding vector by applying the degree-biased random walk and Skip-Gram learning architecture. We further evaluate the performance of our embedding algorithm on the node classification and missing link prediction tasks on various real-world datasets. Compared to state-of-the-art network embedding algorithms, *DiaRW* shows better performance and especially shows high efficiency in large-scale networks with millions of nodes. We apply our algorithm on link prediction task for synthetic networks that have different degree heterogeneity. *DiaRW* performs well with network that has high degree heterogeneity. Chapter 5 provides guidelines for researchers on how to improve the accuracy and efficiency of random-walk-based network embedding algorithms via applying node importance in the random walk process.

Chapter 6 applies the SI spreading model to sample the trajectory paths from a network,

which is further used as the input of the Skip-Gram to derive the embedding of nodes. We apply the SI spreading model to both static networks (*SINE*) and temporal networks (*TSINE*), respectively. The SI-spreading-based algorithm is different from the random-walk-based one, as it captures the tree-like sub-network centered at the seed node. We use the link prediction task to evaluate the algorithms. *SINE* outperforms the static state-of-the-art random-walk-based network embedding algorithms. *TSINE* also performs better than temporal random-walk-based algorithms on temporal networks. Surprisingly, we find that *SINE* performs better than *TSINE* in most of the networks, which indicates that temporal information may not necessarily improve the embedding for missing link prediction. We study the influence of the sampling size, which is quantified by the sum of the length of all the trajectory paths, on the performance of the embedding algorithms. The SI-spreading-based algorithms are less sensitive to the change of the sampling size compared to the random-walk-based ones. Even with a small sampling size, we can obtain a high prediction quality for the SI-spreading-based embedding algorithms. We explain why the SI-spreading-based algorithm can outperform the others by investigating which properties of the local structure make links more likely to be predicted. Even though our SI-spreading-based network algorithms show higher performance than the state-of-the-art embedding algorithms on missing link prediction, link prediction based on simple similarity metrics, such as the number of common neighbors may perform the best in some network datasets. The findings point out the necessity of studying the applicability of different link prediction methods in the different networks.

Both chapters 5 and 6 aim at filling the gap between network science and machine learning algorithms by exploiting the network science theory to design the missing-link prediction algorithms and explain the performance of these algorithms. Considering the node heterogeneity in designing the network embedding algorithms has proven to be both efficient and effective. As a start, we only consider the degree heterogeneity of nodes. The other centrality metrics including the ones that we propose in Chapter 2,3 and 4 can also be applied to this scenario. In addition to the missing link prediction problem and node classification, network embedding algorithms have been applied to other tasks such as future link prediction, clustering and visualization [4–6]. The algorithms we proposed can also be applied to these tasks. For example, we have shown that the temporal information may not necessarily improve the missing prediction performance under SI-spreading-based network embedding. However, the future link prediction task requires to consider the historical evolution of a link to predict its existence in the future. The trajectory paths sampled from the SI-spreading-based network embedding algorithms actually follow the time order of the contacts in temporal networks. In other words, the temporal paths contain the historical evolution of the links and thus may be useful for future link prediction.

7.2. FUTURE WORK

The study of link (node) importance identification in an information diffusion process has successfully followed the following recipe: select the static metrics, define the temporal metrics and evaluate the performance on real-world datasets. The recipe from the network embedding algorithms basically starts from the designing of the embedding algorithms, the selection of state-of-the-art benchmarks, to the explanation of the performance of different algorithms using complex network theory. The two recipes widen our study to other possible directions. Therefore, based on the results and the limitation of this thesis, we raise some promising future directions related to the information diffusion processes.

The role of links (nodes) in other spreading processes. In this thesis, we mainly focus on the spreading models, such as SI and SIR model. The spreading models are proposed to capture different spreading processes. For example, SIS model is used to model how the flu spreads in a population, SEIR model is frequently used to model the spread of the Coronavirus (COVID-19) and the independent cascade model is used to capture the information diffusion process on a social network. Models differ not only in their underlying mechanisms but also in the phenomena they generate. Therefore, it is not trivial to study links (nodes) with what kind of local structure may appear in different spreading processes on a temporal network. Furthermore, different spreading processes simulated on a temporal network may result in different spreading trajectories, such as the random walk and SI model we used in chapters 5 and 6. The study of how to apply other spreading processes to network representation learning is also a promising direction.

Future link prediction in a temporal network. We applied the spreading processes to network representation learning and to solve the missing link prediction problem. Predicting missing links is performed by first hiding some links in a network and then trying to predict them by using the remaining network structure. The future link prediction is different in the sense that we need to consider the historical evolution of a link and even its local structure to predict whether it will appear in the future or not. The difficulty lies in that we need to study the governing process that determine a link's appearance. The temporal network data frequently used nowadays are mainly based on the physical contacts or electronic communications between people. The contacts between people are used for information exchange. Therefore, we believe that spreading processes may play a role in modeling the governing process of the contact formation and thus could be used to predict future links in a temporal network.

Spreading processes for network representation learning on other types of networks. Networks are used to represent different complex systems. For example, bipartite networks are used to represent the user-item interactions in e-commerce systems. The election process can be modeled by opinion model on a signed network. Therefore, the question of how to use the spreading processes for network representation learning on those networks is also a promising topic for future studies.

REFERENCES

[1] D. J. Watts, *A simple model of global cascades on random networks,* Proceedings of the National Academy of Sciences **99**, 5766 (2002).

[2] A. Nematzadeh, E. Ferrara, A. Flammini, and Y.-Y. Ahn, *Optimal network modularity for information diffusion,* Physical review letters **113**, 088701 (2014).

[3] B. Qu, Q. Li, S. Havlin, H. E. Stanley, and H. Wang, *Nonconsensus opinion model on directed networks,* Physical Review E **90**, 052811 (2014).

[4] C. Tu, C. Yang, Z. Liu, and M. Sun, *Network representation learning: an overview,* SCIENTIA SINICA Informationis **47**, 980 (2017).

[5] M. Ou, P. Cui, J. Pei, Z. Zhang, and W. Zhu, *Asymmetric transitivity preserving graph embedding,* in *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD'16 (ACM, New York, NY, USA, 2016) pp. 1105–1114.

[6] S. Cao, W. Lu, and Q. Xu, *Grarep: Learning graph representations with global structural information,* in *Proceedings of the 24th ACM international on conference on information and knowledge management* (ACM, 2015) pp. 891–900.

Acknowledgements

Thank you for everyone who was part of my PhD. With special thanks to Huijuan, Alan and my family.

CURRICULUM VITÆ

Xiuxiu ZHAN

21-10-1989 Born in Anhui, China.

EDUCATION

2008–2012 North University of China, China
B.Sc. in Applied Mathematics

2013–2016 North University of China, China
M.Sc. in Applied Mathematics (Excellent Graduate)
Supervisor: Prof. dr. Z.-K. Zhang
Thesis: The Influence of Information Diffusion on Epidemic Spreading Process

2016–2020 Delft University of Technology, the Netherlands
PhD. in Computer Science
Promotor: Prof. dr. A. Hanjalic
Copromotor: Dr. ir. H. Wang
Thesis: Information Diffusion on Temporal Networks

PROFESSIONAL WORK EXPERIENCE

2014-2016 Joint Training Graduate Student of Computer Science
Hangzhou Normal University
Hangzhou, China

2015 Research Assistant
City University of HongKong
HongKong, China

2018 Research Intern
Tencent
Shenzhen, China

LIST OF PUBLICATIONS

16. **X.-X. Zhan**, Z. Li, N. Masuda, P. Holme and H. Wang, *SI-spreading-based Network Embedding in Static and Temporal Networks*, Submitted to EPJ Data Science.

15. C. Qu, **X.-X. Zhan***, G. Wang, J. Wu and Z.-K. Zhang, *Temporal information gathering process for node ranking in time-varying networks*, Chaos: An Interdisciplinary Journal of Nonlinear Science, **29(3)**, 033116 (2019).

14. Y. Zhang, Z Shi, D Feng and **X.-X. Zhan***, *Degree-biased random walk for large-scale network embedding*, Future Generation Computer Systems, **100**, 198-209 (2019).

13. **X.-X. Zhan**, A. Hanjalic and H. Wang, *Suppressing Information Diffusion via Link Blocking in Temporal Networks*, In International Conference on Complex Networks and Their Applications, Springer, Cham. 448-458 (2019).

12. H. Wang, **X.-X. Zhan**, *Information Diffusion Backbone*, In Temporal Network Theory, Springer, Cham. 199-217 (2019).

11. **X.-X. Zhan**, A. Hanjalic and H. Wang, *Information diffusion backbones in temporal networks*, Scientific Reports **9(1)**, 6798 (2019).

10. C. Liu, N. Zhou, **X.-X. Zhan***, G.-Q. Sun and Z.-K. Zhang. *Information Spreading Dynamics on Adaptive Social Networks*. (Accepted by Applied Mathematics and Computation)

9. **X.-X. Zhan**, C. Liu, G. Zhou, Z.-K. Zhang, G.-Q. Sun, J. J. Zhu and Z. Jin, *Coupling dynamics of epidemic spreading and information diffusion on complex networks*. Applied Mathematics and Computation **332**, 437-448 (2018).

8. **X.-X. Zhan**, C. Liu, Z.-K. Zhang, and G.-Q. Sun, *Epidemic Dynamics On Information-Driven Adaptive Networks.* Chaos, Solitons & Fractals **108**, 196-204 (2018).

7. N. Zhou, **X.-X. Zhan**, Q. Ma, S. Lin, J. Zhang and Z.-K. Zhang, *Identifying spreading sources and influential nodes of hot events on social networks*, In International Conference on Complex Networks and their Applications, Springer, Cham. 946-954 (2017).

6. N. Zhou, **X.-X. Zhan**, S. Lin, S. Yang, C. Liu, Z.-K. Zhang and G.-Q. Sun, *Information Diffusion on Communication Networks*, The Electronic Library **35 (4)**, 745-757 (2017).

5. Z. Ye, **X.-X. Zhan**, Y. Zhou, C. Liu and Z.-K. Zhang, *Identifying vital nodes on temporal networks: An edge-based K-shell decomposition*, 36th Chinese Control Conference (CCC) 1402–1407 (2017).

4. X. Pei, **X.-X. Zhan** and Z. Jin, *Application of pair approximation method for modeling and analysis in the marriage network*, Applied Mathematics and Computation **294**, 280-293 (2017).

3. Z.-K. Zhang, C. Liu, **X.-X. Zhan**, X. Lu, C.-X. Zhang, and Y.-C. Zhang, *Dynamics of Information Diffusion and its Applications on Complex Networks*, Physics Reports **651**, 1-34 (2016)

2. **X.-X. Zhan**, C. Liu, Z.-K. Zhang and G.-Q. Sun, *Roles of edge weights on epidemic spreading dynamics*, Physica A: Statistical Mechanics and its Applications **456**, 228-234 (2016).

1. C. Liu, **X.-X. Zhan**, Z.-K. Zhang, G.-Q. Sun and P. Hui. *How Events Determine Spreading Patterns: Information Transmission via Internal and External Influences on Social Networks*, New Journal of Physics **17(11)**, 113045 (2015).

Propositions

accompanying the dissertation

INFORMATION DIFFUSION ON TEMPORAL NETWORKS

by

Xiuxiu ZHAN

1. The actual time when a link occurs in a temporal network is more important in determining its probability of transmitting information than the number of link occurrences.
   This proposition pertains to this dissertation.
2. When proposing new machine learning algorithms for link prediction, one should always consider the benchmark methods relying on the local network structure.
   This proposition pertains to this dissertation.
3. Increasing the accuracy of predictive methods is less important than understanding their performance.
   This proposition pertains to this dissertation.
4. The more you engineer your life, the more freedom you gain.
5. Imposing a deadline works against the quality of scientific research.
6. The way English is taught in China is not sufficiently supportive for developing the scientific English writing skills.
7. Open public character of social media does not make it an unbiased information channel.
8. Fighting discrimination requires fighting simplistic thinking.
9. Stimulating multidisciplinary scientific approach is the key to innovation in computer science.
10. Online education endangers academic forming at universities.

These propositions are regarded as opposable and defendable, and have been approved as such by the promotor prof. dr. A. Hanjalic.