Information Diffusion Prediction in Social Media

Wang Zhitao

PhD

The Hong Kong Polytechnic University

2020

The Hong Kong Polytechnic University

Department of Computing

Information Diffusion Prediction in Social Media

Wang Zhitao

A thesis submitted in partial fulfilment of the requirements

for the degree of Doctor of Philosophy

November 2019

Certificate of Originality

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgment has been made in the text.

_____(Signed)

Wang Zhitao (Name of student)

Abstract

The emergence of online social media has fundamentally changed the way of information diffusion in human society. These changes trigger a large amount of information diffusion processes in our daily life. Thanks to various social media platforms, the diffusion data become retrievable and traceable, providing unprecedented opportunities as well as great challenges for information diffusion studies. Modeling and predicting information diffusion in social media are meaningful for a variety of real-life applications, such as recommendation services, social marketing campaigns and stability maintenance. In this thesis, we investigate the information diffusion prediction problem from a micro perspective, which aims to model how individuals in social media affect each other in the information propagation process and predict potential individuals who will participate in the future.

Specifically, we identify three key research problems to be solved in information diffusion prediction task, i.e, *1. How to develop an effective prediction model when only historical diffusion processes are observed? 2. How to utilize additional social relationship network to improve the prediction performance and generalization ability of the diffusion model? 3. How to capture interplay effect between information diffusion and social network, and predict diffusion processes and network links jointly?* Inspired by great success of representation learning and neural network techniques on various fields, we propose several models based on the two powerful frameworks to solve the above problems. According to the category of the proposed models, this

thesis is naturally divided into two parts.

In the first part (work 1 and 2), we present two representation learning based models for problem 2 and problem 3, respectively. The poor generalization of discrete graph-based models in previous work motivates us to project diffusion users into a continuous latent space as user representations, which capture unique diffusion characteristics of users. In the representation space, any possible diffusion influence can be flexibly measured by the distance between user representations. In work 1, we focus on problem 2, which asks for integrating social network structure information into diffusion prediction model. In this work, a novel network-regularized role-based representation learning model is proposed. The model learns the user representations based on the objective of maximizing the likelihood of observed diffusion cascades and employs another objective of reconstructing structural proximities as a regularization. The network regularization provides additional constraints on representation learning, correcting the biased information or supplementing the missed information of diffusion relationships. In work 2, we move the attention forward on problem 3, which aims to capture correlations between diffusion cascades and network structure and predict diffusion processes and network links jointly. To achieve this goal, we propose a joint user representation learning model that embeds users as shared representations in a common latent space to characterize their behaviors correspond to both information diffusion and link creation. The proposed model defines two consistent objectives with maximization likelihood estimation on two behaviors and incorporates them in a unified learning framework. The shared representations latently capture the interplay effects and improve the generalization ability on both prediction tasks.

The second part (work 3 and 4) delves into neural network based solutions for problem 2 and problem 1, respectively. Due to the sequential form of diffusion processes, most recent studies formulated diffusion prediction as a sequence prediction

task and employed recurrent neural network (RNN) for the problem. However, few existing RNN-based models consider the observed network structure when modeling diffusion cascades (sequences), which means that they cannot be applied to problem 2. Therefore, in work 3, we propose a novel sequential neural model with structure attention to inject network structure information. The RNN framework is employed to model the sequential information. A structure attention mechanism is designed to capture the important structural information of diffusion users in the given social network. A gating mechanism is further developed to effectively integrate the sequential and structural information. With the injection of structure information, the prediction performance and the generalization ability are further improved. Although retrievable diffusion processes are recorded in the sequential form, we find that non-sequential properties exist, which do not strictly follow the assumptions of previous RNN-based work. In work 4, we propose a hierarchical diffusion attention network with a non-RNN framework for problem 1, which asks to predict diffusion without knowing the underlying social network. The model adopts two-level attention mechanisms, i.e., a dependency attention at user level for capturing historical user-to-user dependencies, and a time-aware influence attention at the cascade (sequence) level for inferring possible future user's dependencies on historical users. The evaluations demonstrate our non-sequential attention network is more effective than previous RNN-based sequential models.

In summary, we present a systemic study of information diffusion prediction in social media. The effectiveness of the proposed models is demonstrated on public benchmark diffusion datasets or synthetic datasets. The proposed models will benefit a wide range of potential applications in real life, such as advertisement recommendation, product influence optimization and personal opinion prediction. Moreover, the possible extensions of current work will provide a deeper and more comprehensive understandings on diffusion related behaviors in social media.

Publications Arising from the Thesis

Journal Papers

1. **Zhitao Wang**, Chengyao Chen and Wenjie Li. Information Diffusion Prediction with Network Regularized Role-based User Representation Learning. ACM Transactions on Knowledge Discovery from Data. 2019. (TKDD)

2. **Zhitao Wang**, Chengyao Chen and Wenjie Li. Joint Learning of User Representation with Diffusion Sequence and Network Structure. IEEE Transactions on Knowledge and Data Engineering. 2019. (TKDE, Under Review)

3. **Zhitao Wang** and Wenjie Li. Neighborhood Attention Networks with Adversarial Learning for Link Prediction. IEEE Transactions on Neural Networks and Learning Systems. 2019. (TNNLS, Under Review)

4. Chengyao Chen, **Zhitao Wang**, and Wenjie Li. Tracking Dynamics of Opinion Behaviors with a Content-Based Sequential Opinion Influence Model. IEEE Transactions on Affective Computing. 2018. (TAC)

Conference Papers

5. **Zhitao Wang**, Yu Lei and Wenjie Li. Neighborhood Interaction Attention Network for Link Prediction. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management. 2019. (CIKM 2019)

6. **Zhitao Wang** and Wenjie Li. Hierarchical Diffusion Attention Network. In Proceedings of the 27th International Joint Conference on Artificial Intelligence. 2019. (IJCAI 2019)

7. **Zhitao Wang**, Chengyao Chen, Ke Zhang, Yu Lei and Wenjie Li. Variational Recurrent Model for Session-based Recommendation. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management. 2018. (CIKM 2018)

8. **Zhitao Wang***, Chengyao Chen* and Wenjie Li. A Sequential Neural Information Diffusion Model with Structure Attention. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management. 2018. (CIKM 2018, * Equal Contribution)

9. **Zhitao Wang**, Chengyao Chen and Wenjie Li. Attention Network for Information Diffusion Prediction. In Companion Proceedings of the The Web Conference 2018. (WWW 2018)

10. **Zhitao Wang**, Chengyao Chen and Wenjie Li. Predictive Network Representation Learning for Link Prediction. In Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2017. (SIGIR 2017)

11. Yu Lei, **Zhitao Wang** and Wenjie Li. Social Attentive Deep Q-network for Recommendation. In Proceedings of the 42th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2019. (SIGIR 2019).

12. Chengyao Chen*, **Zhitao Wang***, Wenjie Li and Xu Sun. Modeling Scientific Influence for Research Trending Topic Prediction. In Thirty-Second AAAI

Conference on Artificial Intelligence. 2018. (AAAI 2018, * Equal Contribution)

13. Chengyao Chen, **Zhitao Wang** and Wenjie Li. Modeling Opinion Influence with User Dual Identity. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. 2017. (CIKM 2017)

14. Chengyao Chen, **Zhitao Wang**, Yu Lei and Wenjie Li. Content-Based Influence Modeling for Opinion Behavior Prediction. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics. 2016. (COLING 2017)

Book Chapters

15. **Zhitao Wang**, Chengyao Chen and Wenjie Li. Information Diffusion Prediction Based on Social Representation Learning with Group Influence. To appear in The Routledge Handbook of Applied Chinese Linguistics. Routledge.

Acknowledgements

I would like to express my deep gratitude to my supervisor, Prof. Li Wenjie, Maggie. Without her patient guidance, continuous support, enthusiastic encouragement and useful critiques on my research work, this thesis would not have been possibly completed. Academically, I learned essential qualities of a good researcher from her. Personally, I received helpful advices and assistances on life issues from her. It is my great honor to be her student.

I am very grateful to all my lab mates, Chen Chengyao, Lei Yu, Cao Ziqiang, Li Yanran, Chen Qiang and etc, for their inspiring discussions, constructive suggestions and kind help on my work. I feel lucky that I could do research in this friendly and talented group. I would also like to thank my office mates and roommate, Cai Sijia, Zhao Ruohan, Xiao Jin, Li Minglei and etc, who enriched my experience and gave me strong support in the Ph.D. life. My grateful thanks are also extended to my remote friends for their continuous concern.

Last but not the least, I would like to express my very great appreciation to my parents, who always give me unconditional support with their love and optimistic life attitude. And most of all, I would like to express the deepest thank and love to my dear girlfriend, who accompany me, support me and love me not only in this Ph.D. journey but also in our life journey.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Information diffusion can be defined as the process that information propagates among people in the society. There is a wide range of information diffusion examples, including the spread of rumors, beliefs, and behaviors. Historically, most studies in this area have been done through field observations and/or phone surveys over a sampled population, focusing on the traditional types of diffusions, which can be summarized as the "word-of-mouth"[20]. However, these studies were limited by the difficulties of data acquisition.

With a drastic boost of users, online social media is gradually taking the place of traditional media, with bringing fundamental changes to the style of information diffusion. Due to these changes, information in social media propagates quickly and evolves dynamically, which produces a large amount of diffusion data with different new-born patterns and characteristics. Understanding these patterns and characteristics of information diffusion in social media has a great significance for individuals to enjoy better social experience (e.g. friends and contents recommendation), for companies to optimize business performance (e.g. social marketing campaigns), for governments to maintain social stability (e.g. terror attack prevention). Therefore, extracting valuable information, e.g., events, interests and knowledge, understanding

meaningful diffusion patterns and predicting temporal propagation in social media are urgently expected by the whole society. Additionally, social media makes the explicit diffusion data much more accessible and inexpensive than before. This offers unprecedented opportunities for the study of information diffusion phenomenon. Therefore, various research topics related to information diffusion, such as popular information (content) detection and influence maximization, have been increasingly explored in social media [58]. Among these topics, information diffusion prediction is one of the most crucial and fundamental problems. It aims at studying how users participate and affect the temporal propagation of information through interactions with other users, such as sharing contents on Facebook or retweeting tweets on Twitter. It should learn mechanisms underpinning information propagation from observed diffusion processes within a diffusion network formed by social relationships among users and predict future diffusion dynamics based on the learned knowledge.

Generally, researches on information diffusion prediction involve two main data sources, i.e., diffusion cascades and diffusion networks. As shown in Fig.1.1, a diffusion network describes the structure of user connections (e.g., following-ships in Twitter and friendship in Facebook) underlying the information diffusion. Diffusion networks provide possible channels for information diffusion flows, thus its structure has a strong correlation with the dynamics of diffusion. Diffusion cascades record how diffusion processes unfold on the network structure, which trigger large amount of user interactions (e.g., retweeting, sharing or discussing). As an effect of these interactions in the diffusion processes, users will change their connections to others to satisfy their interests or social aims.

Although diffusion data merged in social media provide rich sources for researchers, observations on both diffusion processes and diffusion networks are often limited. This brings many challenging problems to the research in this area. As shown in Fig.1.1, on the one hand, observations on diffusion cascades in social media are often

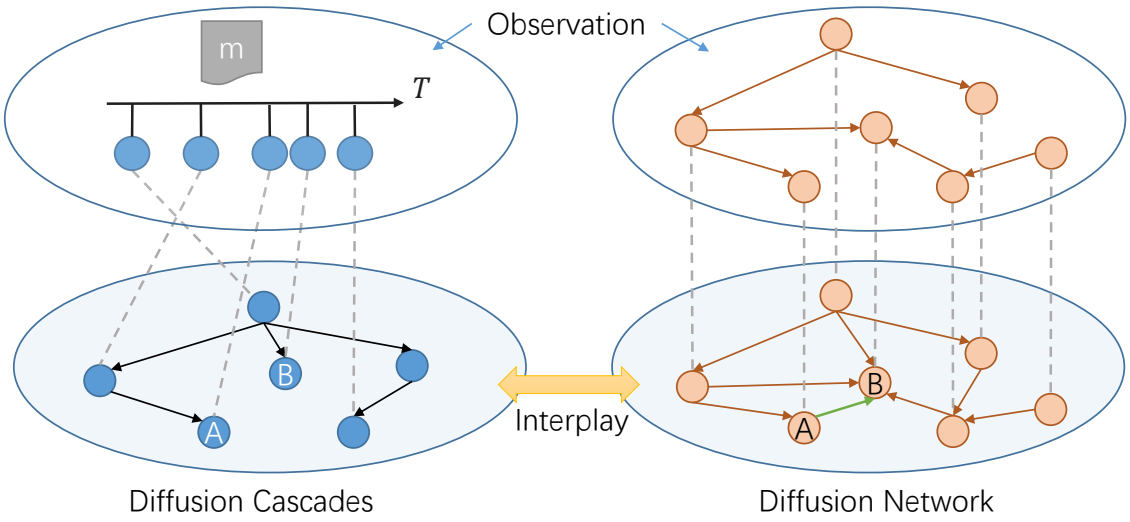


Figure 1.1: Information Diffusion Components in Social Media

actions traces, which only record who participates the diffusion process at which time. We can only obtain diffusion cascades in sequential form but cannot observe *how* and *why* the information propagates among users. On the other hand, we cannot obtain complete diffusion network due to large size of the social network as well as dynamics of user relationships. Therefore, it requires for research approaches with better generalization ability to handle these limitations.

1.2 Research Problems

The major theme of the thesis is to investigate information diffusion prediction approaches, which are able to explain the diffusion mechanisms from observations of diffusion data. And these approaches are expected to tackle the above-mentioned challenging problem, provide insights of diffusion phenomenon, and be applied in real applications. The main research problems in the thesis are listed as follows:

- *Problem 1: How do we develop an effective model that reveals the diffusion mechanism and predicts future diffusion processes when only historical diffusion processes are observed?*

- *Problem 2: How do we improve the prediction performance and generalization ability of the diffusion prediction model if additional social relationship network is observed?*

- *Problem 3: How do we capture the interplay between information diffusion and network structure and predict diffusion processes and unobserved network structure simultaneously?*

The relationships of the three research problems are illustrate in Fig.1.2.

- The first problem focuses only on diffusion processes and requires for the proposal of a predictive diffusion model, which learns the mechanism to explain the observed cascades and has the ability to predict the generation of unobserved or future cascades. The main challenge of this problem is how to learn diffusion mechanism and predict diffusion processes without knowing the explicit user connecting relationships. The prediction model should have the ability to capture user dependencies in sequential diffusion cascades.

- The second problem pays attention to introducing the other component, i.e., diffusion network, for diffusion prediction. The observed network structure is treated as auxiliary information in the prediction model. Network structure provides diffusion relationships, which may not be observed or cannot be inferred from diffusion cascades data. Therefore, incorporating network structure information is expected to improve the generalization ability of diffusion prediction model. The main challenge of this problem is how to effectively integrate network information as additional supervision or constraints in the diffusion model learning.

- The last problem emphasizes on the interplay between diffusion cascades and network structure and expects for a unified model to predict the diffusion and

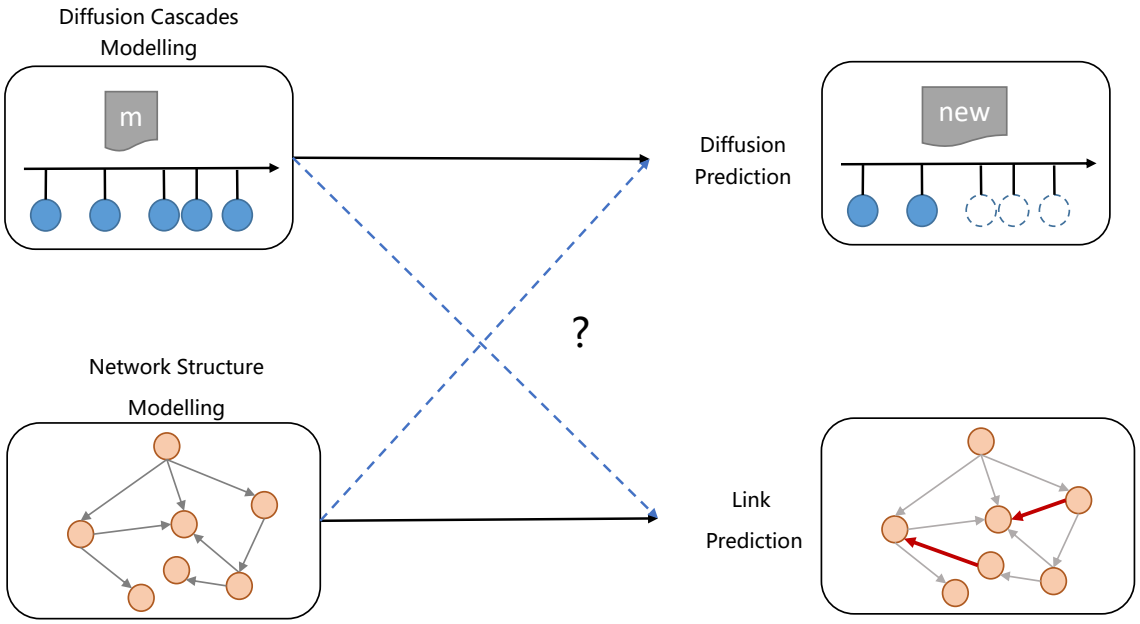


Figure 1.2: Research Questions of Information Diffusion Modeling

network structure jointly. Different from the second problem, which only considers the impact of network structure on diffusion cascades, this problem aims at exploring the mutual impacts between the two signals. The main challenge of this problem is how to unify the two data sources with different structures and how to capture the interplay effects for joint prediction tasks.

1.3 Research Overview and Contributions

In the literature, although graph-based diffusion models are classical solutions for diffusion prediction, previous studies have demonstrated their drawbacks on real social media diffusion data. Therefore, recent research trend on this topic tends to develop more flexible non graph-based models instead of classical graph-based models with heavy assumptions. With the great successes of representation learning and neural networks on a variety of research fields, the research community began to show interest on whether these two powerful data modeling frameworks can be

applied to the diffusion prediction problem. Although a few studies were conducted recently, the progress on this trend remains at initial stage and there is still large space for potential researches.

Therefore, in this thesis, we focus on developing diffusion prediction models based on the two powerful frameworks, i.e., **representation learning** and **neural networks**. Accordingly, studies in the thesis are divided into two parts. The first part pays attention to the representation learning based diffusion prediction models, while the second part concentrates on the neural networks based models. The summarized information of research works in this thesis is presented in Table 1.1. As for representation learning part, two models (work 1 and work 2) are developed to solve the above-mentioned research problem 2 and 3, respectively. As for neural networks part, two models (work 3 and work 4) are proposed, which aim to provide solutions for the problem 2 and problem 1, respectively. The brief overview and contributions of these works are summarized as follows.

Table 1.1: Overview of Research Works in This Thesis

Research Work	Method Category	Research Problem	Publication Venue
Work 1: Network Regularized Role-based User Representation Learning	RL	Problem 2	TKDD [144]
Work 2: Joint Representation Learning from Diffusion Cascade and Network Structure	RL	Problem 3	TKDE
Work 3: Sequential Neural Diffusion Model with Structure Attention	NN	Problem 2	CIKM [143]
Work 4: Hierarchical Diffusion Attention Neural Network	NN	Problem 1	IJCAI [147]

RL: Representation Learning; NN: Neural Networks

Work 1: Network Regularized Role-based User Representation Learning

To overcome the poor generalization ability of discrete graph representation in previous studies, we propose to project diffusion users into a continuous latent space as the role-based (sender and receiver) representations, which capture unique diffusion characteristics of users. A novel network-regularized representation learning model is proposed. The model learns the user representations based on a cascade modeling objective that aims at maximizing the likelihood of observed cascades and employs a matrix factorization objective of reconstructing structural proximities as a regularization on representations. The network regularization provides additional constraints on user representation learning, avoiding over-fitting on diffusion cascades. With the help of network regularization, the model performs more robustly on diffusion prediction problem.

Contributions: We propose to represent diffusion users as role-based representations to capture latent user-specific characteristics. We develop a novel network regularized representation learning model to learn role-based user representations. With the constraints of network regularization, the model is more robust. The experiments on three real-life datasets significantly demonstrate the effectiveness of the proposed model on diffusion prediction. This work has been published as a journal paper on ACM Transactions on Knowledge Discovery from Data (TKDD) [144].

Work 2: Joint User Representation Learning from Diffusion Cascade and Network Structure

In the work 1, we only regard network structure as auxiliary information in representation learning model. In fact, there exists interplay effects between diffusion cascades and network structure [149, 6]. In this work, therefore, we aim at exploring the correlated effect of diffusion cascade and network structure. To achieve this goal,

we propose a joint user representation learning model that projects users as shared representation in a common latent space to characterize their behaviors corresponding to both information diffusion and link creation. The proposed model defines two objectives on two behaviors and incorporates them in a unified framework. The shared representations capture the interplay effects of the two behaviors and improve the generalization ability on both diffusion prediction and link prediction.

Contributions: We explore the correlation between information sharing and relationship building behaviors with behavior-shared user representations. We propose a novel joint learning model to learn user representations simultaneously from both diffusion sequence and social network. The proposed user representation learning model is applicable to any correlated behavior modeling and able to tackle other similar tasks. Moreover, it is easy to be extended to cope with multiple diffusion-related behaviors in social media. The experiments on real datasets prove the superiority of the proposed model on both diffusion prediction and link prediction tasks. This work is under the review (minor revision) of IEEE Transactions on Transactions on Knowledge and Data Engineering (TKDE).

Work 3: Sequential Neural Diffusion Model with Structure Attention

Several recurrent neural network (RNN) based models were recently proposed and proved effective on diffusion prediction. However, none of them consider the observed network structure when modeling diffusion cascades (sequences). Therefore, in this work, we propose a novel sequential neural information diffusion model with structure attention (named SNIDSA) to jointly model sequential diffusion cascades and structural social graph for diffusion prediction. The recurrent neural network framework is employed to model the sequential information. A structure attention mechanism is designed to capture the structural dependency among users, which is

defined as the diffusion context of a user. A gating mechanism is further developed to effectively integrate the sequential and structural information. With the injection of structure information, the prediction is further improved.

Contributions: We are the first to propose an attention mechanism to incorporate structure information under the recurrent sequential neural framework. The prediction ability and the robustness of SNIDSA are verified on four synthetic datasets and a real diffusion dataset. This work has been published as a conference paper on ACM International Conference on Information and Knowledge Management. (CIKM'18) [143].

Work 4: Hierarchical Diffusion Attention Network

Although recent sequential RNN-based models were proposed for diffusion prediction, it is found that non-sequential properties exist in real diffusion cascades, which do not strictly follow the sequential assumptions of previous work. In this work, we propose a hierarchical diffusion attention network (HiDAN), which adopts a non-sequential framework and two-level attention mechanisms, for diffusion prediction. At the user level, a dependency attention mechanism is proposed to dynamically capture historical user-to-user dependencies and extract the dependency-aware user information. At the cascade (i.e., sequence) level, a time-aware influence attention is designed to infer possible future user's dependencies on historical users by considering both inherent user importance and time decay effects. The hierarchical attention mechanisms are proved effective to capture non-sequential dependencies in diffusion cascades.

Contributions: To the best of our knowledge, we are the first to propose a non-sequential neural network framework for diffusion prediction problem, which is well-adapted to properties of real diffusion cascades. We design two-level attention mechanisms for cascade modeling, i.e., a user-level dynamic dependency attention for

historical diffusion dependencies, and a cascade-level time-aware influence attention for future dependencies. The experiments on three real datasets demonstrate the significantly improved effectiveness and efficiency of the proposed model compared with state-of-the-art approaches. The initial version and the full version of this work have been published as a poster on the Web Conference 2018 (WWW'18) [142] and a long paper on the 27th International Joint Conference on Artificial Intelligence. (IJCAI'19) [147].

1.4 Structure of Thesis

The overall picture of the thesis is as follows. Chapter 1 briefly introduces the background of information diffusion studies, the main research problems, and the overview of this thesis. Chapter 2 provides a comprehensive survey on the previous research work related to this thesis, including mainstream graph-based and non-graph-based diffusion models, network representation learning methods, link prediction studies and other diffusion-related researches. Chapter 3 presents preliminaries of diffusion data, research problems and related techniques. Afterwards, the thesis is divided into two parts according to different methodologies. The first part (Chapter 4 and 5) mainly introduces the proposed representation learning-based models. Chapter 4 presents a novel representation learning model which introduces network structure as a regularization on user representation learning from diffusion cascades. Chapter 5 investigates the possibility of jointly learning user representation from both diffusion cascades and network structure to simultaneously solve diffusion prediction and link prediction problems. The second part (Chapter 6 and 7) presents the proposed neural networks-based models. Chapter 6 studies how to integrate network structure with attention mechanism into classical recurrent neural network framework. Due to the limitation of recurrent neural network, in Chapter 7, a novel non-sequential

neural architecture is proposed for diffusion prediction when the underlying network is unknown. At last, the final chapter, Chapter 8 summarizes the proposed methods, findings, conclusions, and contributions of the work. The potential extensions of the current work are also suggested.

Chapter 2

Literature Review

In this chapter, we survey the studies related to this thesis. We mainly focus on summarizing mainstream information diffusion prediction methods, which are categorized into graph-based and non-graph-based. We also give a wide review of literatures with closely related topics, i.e., network representation learning, link prediction and other diffusion applications.

2.1 Graph-Based Diffusion Prediction Models

A large number of previous studies held an assumption that information diffusion processes unfold on a graph formed by connections of diffusion nodes (users). The graph can be a social network, a communication network or networks of any other relationships. Edges of the graph are assumed as user-to-user diffusion channels and information can only diffuse along with these edges. The general goal of graph-based models is to infer the diffusion influence on edges of the diffusion graph and predict future diffusion based on the graph with learned influence. In previous studies, diffusion graph can be either explicit or implicit, thus the following review will introduce graph-based models from these two branches.

2.1.1 Explicit Graph Based Models

This branch of studies aims at modeling diffusion process when an explicit diffusion graph is given. Most existing explicit graph-based models were based on two fundamental theoretical models, i.e., Independent Cascade (IC) model [50] and the Linear Threshold (LT) model [55], which have different graphical assumptions on diffusion process. In the case of IC assumptions, users can be activated independently by one of their parents. In the case of LT assumptions, the infections of users depend on the aggregated influence of activated parents.

Some concepts of the basic IC and LT models are introduced as follows. For both models, a node once activated will stay activated. The diffusion processes unfold at discrete time step and the processes continue until no more activations occur. For the IC model, given a set of initially activated nodes, at each time step, each activated node tries to influence one of its inactive child (neighbor). Regardless of its success, the same node will never get another chance to activate the same inactive child (neighbor). The success of node u in activating the child (neighboring) node v depends on the independent probability of the edge (u, v) defined as $p_{u,v}$. For the LT model, each node v is associated with a threshold $t_v \in (0, 1]$, drawn from some probability distribution. Every parent (neighboring) node u of v has a non-negative weight $w_{u,v}$. Given a set of initially activated nodes, at each time step, each inactive node v will be activated if the overall influence of its activated parents exceeds its threshold. The IC model and the LT model can be unified under a more general framework like the General Threshold and Cascade Models [67].

Many variants were developed to relax or change the constraints and assumptions on diffusion processes in the original IC and LT models. For instance, Saito et al. [114, 115] firstly developed continuous-time versions of IC and LT models (CTIC and CTLT) to relax the constraint that diffusion unfolds with discrete time steps.

Du et al. [39] then proposed a more efficient algorithm to estimate parameters in CTIC model. Afterwards, the asynchronous versions of IC and LT (AsIC and AsLT) were proposed in the work [117], which relaxed the synchronicity constraint. The AsIC and AsLT models proceeded iteratively along a continuous time axis and an additional time-delay parameter on each edge of the graph in addition to parameters in the original versions. Another model [116] proposed a different assumption that a node can be activated multiple times instead of only once since it is reasonable that an activated user diffuse a same piece of message several times. In the work [53, 68], the diffusion influence of a node on another node was assumed to have an exponential decay over time and a node can consider not only the influence of activations happened one time step before but also that of the earlier ones. Another work [75] assumed that nodes can deactivate from active states and adopted an additional deactivation threshold in the LT to model non-monotonic behavior.

There were other IC-based and LT-based models focusing on integrating rich features of either nodes (users) or diffused messages. In terms of using node features, Saito et al. [119] proposed to introduce available user attributes into IC. Yang et al. [156] studied how social roles of users affected the diffusion influence. They proposed a role-aware information diffusion model that integrates social role recognition and diffusion modeling into an IC-based generative model. In terms of using content features, Galuba et al. [47] proposed a LT-based model with considering the virality features of diffused information to predict retweeting behavior. The work [13] tried to capture topics in the diffused content and proposed Topic-aware IC and LT models, which considered that views and opinions under different topics diffuse differently. Besides, some other work, such as [78], developed comprehensive models by using both features of diffused contents and users.

A major limitation of above-mentioned models is that they cannot handle the situation when the observed explicit network is incomplete [134] and dynamic [121].

Therefore, some recently proposed explicit graph-based models attempted to predict the diffusion process with partial structure information. For instance, Duong et al. [42] studied the problem of modeling information diffusion when the network is only partially observed. They inferred missing edges indirectly or directly and predicted diffusion based on the more complete graph.

2.1.2 Implicit Graph Based Models

This branch of researches assumes that the diffusion network is implicit. The aim is to infer the underlying diffusion network that explains the observed sequential cascades without any prior graph information. These models make it possible to retrace the path taken by a piece of information only from observation of sequential diffusion process. They were not developed specifically for diffusion prediction task, but they were flexibly used for predicting diffusion with the inferred graph.

Initial researches in this branch focused on inferring network connectivity only. All these methods assumed that the pairwise transmission model was fixed with a predefined transmission rate. Gomez et al. [51] proposed the NetInf algorithm, which explored correlations in nodes infections times to infer the structure of the diffusion network with IC assumption. The NetInf utilized an iterative algorithm based on submodular function optimization for discovering diffusion graph. Another model called MULTITREE [111] also formulated the network inference problem as a submodular maximization problem and considered all directed trees of each cascade. Myers et al. [102] proposed CONNIE model which presented a maximum likelihood approach based on convex programming with a l_1-like penalty term to encourage sparsity of the inferred network.

Apart from inferring network connectivity, most previous models aimed at estimating pair-wise transmission rates between users. A general framework named NetRate was firstly proposed based on survival analysis theory [81, 80] in the work

[112], where the likelihood of a node infecting another at a given time was modeled via a probability density function depending on infection times and the transmission rate between the two nodes. NetRate formulated the diffusion network inference as a convex optimization problem. Based on this framework, many models were developed by considering different factors from different views. The work [52] extended NetRate to dynamic version called InfoPath, which used stochastic gradients to provide online estimates of the structure and temporal dynamics of a network. The model TOP-ICCASCADE proposed in [41] aimed to infer not only the hidden diffusion networks but also the topic dependent transmission rates from the observed time stamps and contents of cascades. Wang et al. [137] investigated the effects of cascades patterns with NetRate framework and provided an expectation maximization algorithm to effectively infer hidden network. Du et al. [40] argued that the diffusion influence between users is heterogeneous, which cannot be described by a simple parametric model, and proposed a kernel based method which can capture a diverse range of different types of influence. Another following model CENI [62] integrated clustering strategies to NetRate to improve the efficiency of network inference. The work [1, 36] then presented very detailed theoretic analysis of these pair-wise transmission rates learning models, such as the recovery condition and the sampling complexity. Apart from leveraging the above-mentioned survival analysis theory, some studies adopted other techniques in stochastic process field. For example, Iwata et al. [64] proposed to model the diffusion cascades with Poisson process and developed a Bayesian inference algorithm. All above learning models relied on some parametric functions to describe the relationship between pair-wise transmission rates and infection time. Recently, Rong et al. [113] argued that complex diffusion process in the real world is hard to be captured by a parametric model and proposed an algorithm named NPDC to interpret the diffusion process in a non-parametric way. This algorithm inferred the diffusion network according to the statistical difference of the infection

time intervals between nodes connected with diffusion edges versus those with no diffusion edges, which did not need definitions of any transmission models between nodes.

Additionally, some recent work attempted to infer other forms of implicit diffusion structures to explain the observed diffusion processes. For instance, Kurashima et al. [76] proposed a probabilistic model for inferring the diffusion network in the continuous space. The latent coordinates of users in the continuous space explained the diffusion relationships extracted from observed cascades, providing insights into analyzing diffusion behavior of different user groups, and led to high accuracy in inferring the underlying network when analyzing the diffusion process of new or rare information. Bao et al. [12] proposed to explain the diffusion process with "motif", which postulated that the latent temporal activation motifs (patterns) of different social roles form the underlying information diffusion mechanisms generating the information cascades. They formulated the inference of the temporal activation motifs as a probabilistic problem and developed an EM algorithm to infer the diffusion-specific motifs with the diffusion probabilities.

2.2 Non-Graph Based Diffusion Prediction Models

Apart from above mentioned graph-based methods, numerous researches proposed to model information diffusion with other dynamic mechanisms instead of explaining how diffusion cascades unfold along with graph structure. Despite of not using graph assumptions, network information is still utilized by non-graph-based models in different manners. We summarize this kind of studies as non-graph-based diffusion models, which are categorized into **macroscopic models** and **microscopic models**.

2.2.1 Macroscopic Non-Graph Based Models

The majority of non-graph-based studies attempted to explain diffusion processes from a macro perspective. The general goal of these studies is to predict dynamics of diffusion cascade size (popularity) instead of individual infections. We will introduce four groups of macro non-graph-based diffusion models, i.e., epidemic models, feature-based models, stochastic process models and neural models.

Epidemic Models: Before the emergence of social media, information diffusion on real-life social networks has been already studied based on epidemiology theory in the social, physical and computational sciences for a long period. Therefore, early macro non-graph-based models were developed referring to the epidemiology theory. In the epidemiology theory, information entities are regarded as infectious diseases and the diffusion process is analogy to contagion process through social connections. There are two seminal models in epidemiology, i.e., SIS [9] model and SIR [5] model, where S stands for "susceptible", I for "infectious" (i.e. adopted the information) and R for recovered (i.e. refractory). The basic theory is that the infected population grows exponentially until the rate of infection is balanced by the rate of recovery, or the contagion finally dies off when the recovery rate prevails. Neither the SIS nor SIR model considers the underlying network structure of how individuals are connected. Based on SIS, Leskovec et al. [85] proposed a simple and intuitive model that requires a single parameter β. It assumes that all nodes have the same probability β to adopt the information and nodes that have adopted the information become susceptible at the next time-step. This is a strong assumption since in real-world social networks, influence is not evenly distributed between all nodes. Yang et al. [154] defined an influence function that quantifies how many subsequent infections can be attributed to the influence of that node over time rather than a simple diffusion parameter for each node. They developed a Linear Influence Model to estimate the influence

function.

Stochastic Process Models: Recently, another family of macro non-graph based models was developed based on the theory of stochastic process. These methods successfully captured the "rich-get-richer" social phenomena in diffusion processes and are able to output an estimation of the final size or incremental size of a diffusion cascade. In the seminal work, Crane et al. [34] firstly showed how to model the diffusion bursts and decays based on a Hawkes point-process. Afterwards, more complex models have been proposed to predict diffusion popularity in microblogs [48] and videos [37]. To enhance previous models, Shen et al. [122] employed reinforced Poisson processes, modeling three phenomena: fitness of an item, a temporal relaxation function and a reinforcement mechanism. Another crucial work SEISMIC was proposed by Zhao et al. [160], who extended the previous point process-based models by employing a double stochastic process, one for infectiousness and the other one for the arrival time of events. In this way, they allowed the infectiousness of information to change over time. Mishra et al. [101] proposed a simpler and more intuitive generative point process model with the same information used in SEISMIC. Some researchers [155, 45] also utilized the multivariate Hawkes Processes model to tackle both diffusion popularity prediction and network dynamics prediction.

Feature-based Models: This group of methods heavily depends on feature engineering. A set of potentially relevant features, such as content features, network structural features, temporal features and user features, were firstly designed and extracted from diffusion data. Then different learning algorithms, e.g., linear regression [29], regression trees [10] and passive-aggressive algorithms [108] were applied based on the extracted features to predict the dynamic of diffusion cascade size.

Neural Models: The above feature-based methods need large amount of effort on feature engineering since their performance is highly sensitive to the quality of the features [11]. Recently, the feature learning abilities of neural networks have been

demonstrated in various tasks. This provides a potential way to break the limitations of traditional feature-based method. Li et al. [87] proposed a neural model named DeepCas to predict incremental size of diffusion cascades. By using techniques of random walk and recurrent neural network, DeepCas embedded structure information of diffusion cascades as deep features. Based on the constructed features, a multilayer perceptron regressor was used for prediction. Cao et al. [22] proposed another neural model DeepHawke. Compared with DeepCas, DeepHawke additionally designed a neural Hawkes point-process function to estimate the non-linear time-decay effects.

2.2.2 Microscopic Non-Graph Based Models

Recently, a series of non-graph-based models attempted to explain and predict information diffusion from a micro view without adopting graph assumptions in previous graph-based models. Instead of inferring interpersonal influence, these models aimed to capture diffusion interactions in different forms. These state-of-the-art studies provide very innovative manners for solving diffusion prediction problem. The studies in this thesis, therefore, mainly focus on this trending direction. The state-of-the-art models in this direction can be categorized into two groups, i.e., representation learning based models and neural network-based models.

Representation Learning Models: Representation learning has been recognized a strong problem-solving technique in many tasks. With the success of representation learning methods in various fields, a group of recently proposed representation learning models aimed to learn user features in a continuous latent space instead of inferring interpersonal influence on the discrete graph. In the latent space, all possible pairwise diffusion influence could be measured by the representation distance, providing a more global and unbiased manner of modeling diffusion relationships. Since the learned user representations are low-dimensional vectors, i.e., smaller than total number of distinct users, representation learning models have fewer parame-

ters compared with previous implicit graph-based models, which remarkably reduces computation complexity. The first representative diffusion representation learning model was proposed in [17], which interpreted diffusion cascades as ranking lists and learned representations by optimizing a simplified ranking problem. The results of this work witnessed the good performance of latent representations for diffusion modeling. However, the idea of ranking diffusion users with heat diffusion kernels [77] assuming all users receive information from the single source, had conflicts with real diffusion properties. Recently, Bourigault et al. [18] proposed another representation learning model, which held graph-free IC-like assumptions to extract more robust diffusion probabilities with user representations. The model showed better generalization ability on various real-world datasets. This work provided insights for integrating representation learning with the classical diffusion theory. A very similar work [76] adopted a general framework NetRate [112], which was originally proposed for diffusion network inference, to learn robust user representations. Another recent work [138] attempted to capture context-dependent factors like cumulative effect in the representation learning model. To the best of our knowledge, the development of representation learning diffusion models is still at initial stage. All existing models learned representations singly from diffusion sequences and were not flexible to integrate any available structure information of user graph. These drawbacks also made them fail to explore the correlations between diffusion dynamics and structure dynamics. Therefore, in this thesis, we aim to develop more generalized and robust representation learning models, which are able to encode diffusion and structure information in the representation space and predict dynamics in both types of information jointly.

Neural Models: Since diffusion cascades are recorded in the form of sequence, latest studies began to transform diffusion prediction problem as a sequence prediction and attempted to solve it with sequential models. Inspired by the outstanding

sequence modeling performance of recurrent neural network (RNN), several state-of-the-art RNN-based models for information diffusion were developed. RNN based models sequentially encode historical infection information as hidden states with recurrent neural layers and predict next infected user at each time step based on the encoded states. RMTPP proposed in [38] was the first RNN-based model for predicting cascade dynamics. It not only adopted RNN framework to model diffusion user sequence but also designed a neural point process function to capture temporal effects of users in the diffusion processes. This model demonstrated the effectiveness of RNN on diffusion prediction problem and built a foundation for following proposed neural models. Another representative work is CYANRNN [139]. They extended the RMTPP model to an encoder-decoder framework, which employed a popular machine translation alignment mechanism. However, this model did not show good performance in our experiments. The encoder-decoder framework, which was originally designed for sequence-to-sequence problem, may be over-complicated for this single-sequence modeling problem and the used alignment mechanism may not clearly explain the interpersonal diffusion dependency. The Topo-LSTM proposed in [136] employed the long short-term memory (LSTM) [61] recurrent framework instead of vanilla RNN architecture in RMTPP. It also integrated the structure information. The gating mechanism in LSTM improved the performance. However, over-depending on the graph structure in sequence encoding limits its application when the graph is unknown or partially observed. Therefore, in this thesis, we firstly study how to integrate partial graph as auxiliary information instead of over-depending on it in recurrent framework. More importantly, we identify an important characteristic of diffusion cascade that users in cascades are not sequentially dependent, which means that sequential assumptions in above mentioned models may not stand any more. To this end, we focus on developing non-sequential neural models for diffusion prediction, which could be well-adapted to the unique properties of

diffusion sequences.

2.3 Network Representation Learning

The diffusion network consisting of user connections plays a crucial role in information diffusion modeling. Most previous information diffusion work heavily focused on developing models under the graph-based framework or manually extracting features from network structure. The recent research attentions on network representation learning enable us to learn and leverage informative structural features of users more effectively and flexibly. Therefore, integrating network representation learning into information diffusion modeling is a very potential way to improve the current models, but there are very limited studies on this direction. The following contents will present a survey of state-of-the-art researches on network representation learning.

The aim of modern network representation learning methods is to represent nodes as vectors in a continuous space (a.k.a., node embeddings) so that nodes with similar embedding vectors share structural proximities. The existing methods can be categorized into two groups: matrix factorization and neural embeddings.

Matrix Factorization: In matrix factorization, a network is represented as a matrix where the entries represent relationships (for instance, relationship strength, edge weight, or frequency of communication). Then the matrix is projected to a low dimensional space using linear techniques based on matrix factorization approach, such as SVD or PCA on the covariance matrix [3, 128]. Alternatively, the matrix can be embedded into the latent continuous space by using non-linear techniques such as multi-dimensional scaling [33] or IsoMap [130]. These methods cannot avoid the drawbacks, such as expensive computation for large real-world networks and ignorance of important network properties, e.g., sparsity and power law distributions. A series of recent work [153, 23, 152] focused on translating the neural embedding

as equivalent matrix factorization models. Yang et al. [152] presented a mathematical proof of the equivalence between neural methods and their proposed matrix factorization model. Afterwards, this factorization model is extended for the specific tasks. For example, Yang et al. [153] integrated the text information to improve the classification performance. Tu et al. [35] added a max-margin classifier in the framework to enhance the discrimination ability of learned representations. Cao et al. [23] developed a more detailed matrix factorization framework, which preserved both local and global structural information of network. A very recent work [110] proved that most existing network embedding models can be unified into the matrix factorization framework, which provided a flexible matrix factorization-based tool.

Neural Embeddings: Inspired by the huge success of deep learning applied to various domains such as text [100] and image [74], a series of neural models were developed for network representation learning. Perozzi et al. [107] was the first to propose a representative neural model DeepWalk, which made an analogy between the nodes in networks and the words in natural language. They used fixed-length random walk paths (node sequences) to simulate the sentences (word sequences) in natural language so that node representations can be learned using the word representations learning methods, e.g., SkipGram [100]. Based on the SkipGram, Tang et al., [127] proposed another network representation learning model to embed the first order and second order proximities of nodes separately. They utilized the negative-sampling strategy for optimization, which was different from DeepWalk. Grover et al. [56] investigated the sequence sampling strategies under DeepWalk framework and presented a very detailed sampling model. Based on above models, a series of extensions with considering different information has been proposed. For example, Li et al. [90] leveraged label information of nodes to enhance classification ability. Other work attempted to integrate the group information [26] or community information [131, 157] to improve the community detection performance of learned

representations.

Apart from the above shallow neural network models, some recent work paid attention to develop the deep architecture for network representation learning. Wang et al. [135] developed a semi-supervised deep model to jointly preserve the first-order and second-order proximity of nodes. Chang et al. [25] focused on the representation learning problem for heterogeneous networks and tackled this problem by designing a new deep architecture. The work [24] employed a random surfing model to capture graph structural information as a data matrix directly instead of sampling and designed a deep neural network to learn the representations from this data matrix. Meanwhile, a series of studies focused on designing graph-specific neural architecture or neural layer for an end-to-end learning, where the features learned in hidden layers in these architectures were regarded as deep node representations. Niepert et al. [105] designed a convolutional neural network-based framework for arbitrary graphs. Analogous to image-based convolutional networks that operate on locally connected regions of the input, they presented a general approach to extracting locally connected regions from graphs. Kipf et al. [71] proposed a representative graph convolutional network (GCN). They designed a new graph convolutional operation, i.e., a filter as approximate first-order Chebyshev polynomials of the diagonal matrix of eigenvalues. Recently, inspired by attention mechanisms in other tasks, a graph attention network [133] was proposed. The operation of graph attention layer was aggregating the features of neighbors as hidden features of the center node. The attention mechanism was employed to determine the weights of a node's neighbors when aggregating feature information.

Another line of related work is the representation learning of other structures in graphs rather than individual node. For example, Zheng et al. [162] studied community representation learning problem. They regarded community embedding as providing a higher-order proximity to define the node closeness and proposed

ComEmbed to jointly optimize the community embedding and node embedding. Many other studies were based on graph kernels, measuring pairwise similarities between graphs. For example, the Weisfeiler-Lehman subtree kernel (WL) [124] computes the graph similarity based on the sub-trees in each graph. Meanwhile, some deep neural network models have been proposed to improve the effectiveness of graph kernels learning. Yanardag et al. [151] presented a unified deep framework to learn latent representations of sub-structures for graphs, which leverages the dependency information between sub-structures by learning their latent representations. The subgraph2vec approach proposed in the work [103] was for rooted subgraphs representation learning. This approach leverages local information obtained from neighborhoods of nodes to learn the latent representations in an unsupervised way.

2.4 Link Prediction

Apart from dynamics of diffusion information, network structure dynamics also exist and take important effects in social media. Predicting missing or promising links on networks is of great importance. For example, inferring potential friendships enhances user experience in online social media services [2]. Predicting edges on item-user graph improves the recommendation accuracy in e-commerce network [73]. Reconstructing links help to complete information in knowledge graph [104]. Identifying the latent connections on protein-protein network saves large amount of human effort on blindly checking [4]. Link prediction has attracted increasing attention from researchers in recent years [95, 98]. Existing link prediction approaches can be categorized into three families: heuristic methods, feature based methods and neural network-based methods.

Heuristic Methods: Most heuristic methods calculate a similarity score of a node pair based on different structural information. Some studies defined similarity scores

based on the neighborhood structural information. Popular neighborhood-based heuristics include first-order methods Common Neighbors, Jaccard Index [120] and Preferential Attachment [91]; second-order methods, i.e., Adamic-Adar [2], Resource Allocation [165]; and high-order heuristic SimRank [65]. Some other studies computed similarity based on paths between a node pair. Representative path-based heuristics include Katz Index [66], Local Random Walk [94] and Random Walk with Restart [19]. Though simple, these methods can still achieve competitive performance compared with other methods.

Feature Based Methods: The idea of feature based methods is designing or learning a set of features for node pairs and train a link classifier based on the feature set. A traditional way was designing features based on above mentioned heuristics. In recent studies, several latent feature learning models were proposed and used for link prediction. The most classical one is the matrix factorization method proposed in the work [99]. This method learned latent features (embeddings) of nodes by factorizing adjacency matrix of the network, and link scores of node pairs were predicted with inner product. Besides, the previously introduced network representation learning models, such as DeepWalk [107], LINE [127] and Node2vec [56], are also applicable for link prediction. Grover et al. proposed several composition operations in the work [56] to construct features of node pairs based on the learned representations and trained a classifier. The experiments showed that latent node features learned by generic network representation learning models were good at predicting links. In a very recent work [141], Wang et al. proposed the PNRL model, which was a link prediction-specific network representation learning model. This model learned node embeddings by simultaneously preserving proximities of observed structure and inferring hidden links, thus the learned embeddings were more predictive compared with previous generic network embedding models.

Neural Methods: Recently, some neural network-based link prediction models

were developed, which explored non-linear deep structural features with neural layers. These models allowed to solve link prediction problem in an end-to-end manner. Here we introduce some representative methods. Variational graph auto-encoder proposed in the work [72] employed the graph convolutional network (GCN) [71] to encode the graph adjacency matrix and reconstructed observed links or predicted unobserved links with an inner product decoder. Another state-of-the-art neural model WLNM [159] used a graph labeling algorithm to transform the enclosing subgraph, which was structurally combined neighborhood of the node pairs, as a meaningful matrix. A convolutional neural network (CNN) was then employed to encode the labeled matrix, and link score was predicted via a classification layer. Instead of encoding whole graph or neighborhood, Zhao et al. [161] proposed a path-based neural model. The model employed LSTM to encode possible paths between two nodes and inferred the link score based on the encoded structural path information. Recently, Wang et al. [146] proposed a novel attention neural network, i.e., NIAN. The proposed neighborhood interaction attention mechanism is able to automatically learn comprehensive neighborhood interaction features for link prediction.

2.5 Other Diffusion Related Researches

Diffusion Source Identification

There is a small branch of studies on the diffusion source identification problem. This problem is very important in practice. A representative scenario is the detection of rumor diffusion. Due to wide spreading of rumors, it is impossible to observe the beginning speakers of rumors in a large-scale network. Therefore, identifying the rumor sources from partially observed diffusion processes is of great significance in this scenario. The diffusion source identification has not been studied until recent years. Lappas et al. [79] considered the problem of selecting a set of k active nodes named

"effectors" that best explain the observed activation state, under a given information propagation model. They formally defined the k-effectors problem and used a dynamic programming algorithm to solve it optimally. Prakash et al. [109] proposed to employ the Minimum Description Length (MDL) principle to identify the set of seed nodes and model the virus propagation ripple starting from those nodes that best describes the given snapshot of a partially infected network. Recently, Farajtabar et al. [44] proposed a two-stage scalable framework to tackle the problem in a continuous-time condition. They first learned a continuous-time diffusion network model based on historical diffusion traces and then identified the source of an incomplete diffusion trace by maximizing its likelihood under the learned model.

Influence Maximization

The influence maximization (IM) studies investigated the effects of information diffusion and focused on the problem: which seed users should be chosen in the initial phase of the diffusion in order to maximize the range of the diffusion? Most of the existing IM methods employed a simple greedy algorithm framework. The algorithm is initialized with an empty seed set S, and it iteratively selects a node u into S if u provides the maximum marginal gain to the influence function with respect to seed set. The algorithm terminates when there are k distinct nodes in S. Evaluating influence of a seed set was proved a #P-hard problem even under simple IC or LT diffusion model, which remarkably limits the efficiency of this algorithm framework. According to the way of estimating the influence function of seed set, existing studies can be classified into three categories: simulation-based approaches, proxy-based approaches and sketch-based approaches.

The key idea of simulation-based approaches is to perform Monte-Carlo (MC) simulation for evaluating the influence of seed set. Given the seed set, the approaches conduct diffusion simulation under IC or LT models until no more users are activated.

The seminal work [67] used a naive MC simulation, which was prohibitively expensive against large graphs. The follow-up studies focused on reducing the complexity of multi-round MC. Some methods aimed to reduce the rounds of MC. For example, the methods CELF [84], CELF++ [54] and UBLF [164] were proposed to estimate the upper bound of the influence functions in order to prune the ones with insignificant influences. Some other algorithms attempted to reduce the complexity of individual round in MC simulation, such as the CGA algorithm [140].

The idea of proxy-based approaches is to design proxies instead of running heavy MC simulations to approximate influence function. Although some structural ranking proxies, e.g, PageRank [106] and Distance Centrality [46], could be applied to select the seeds, they failed to consider the diffusion mechanism of influence. Therefore, some proxy-based approaches attempted to design diffusion model-based proxies. For example, Kimura et al. [69] proposed SPM and SP1M models to reduce the stochastic IC model to a deterministic model where the influence spread of any seed set can be computed exactly. A series of studies [27, 27] restricted the influence range of each user in IC or LT to a small local subgraph. The main problem of proxy-based methods was the lack of approximation guarantee.

The main focus of sketch-based approaches is to improve efficiency of the simulation-based methods while preserving the approximation guarantee. To avoid rerunning the MC simulations, the sketch-based approach precomputed a number of sketches based on the specific diffusion model, and then exploited the sketches for evaluating influence spread. Representative algorithms include forward influence sketch methods [28, 30] and reverse reachable sketch methods [16, 129].

Chapter 3

Preliminaries

In this chapter, we will introduce preliminaries about the diffusion data, research problems and related techniques.

3.1 Diffusion Data

3.1.1 Diffusion Cascade

Formally, an observed diffusion process of a specific piece of information is denoted as a diffusion cascade $c = \{(u_1, t_1), ..., (u_i, t_i), ...(u_n, t_n)\}$ with a sequential form. The element (u_i, t_i) indicates that u_i is the ith user who propagated this piece of information at time t_i. The participation of u_i in cascade c is often called as u_i is "infected" by cascade c at time t_i. The elements are ordered by their infection time, thus $t_i < t_j$ if $i < j$. Other users, who are not infected by this cascade, are often called as "survival" users of cascade c.

3.1.2 User Network (Graph)

Users in social media are often connected with some relationships, such as following connections in Twitter or citation links between blogs. Due to these connections, diffusion users naturally form a user network. A network with N users is defined as $G = (U, E)$, where each vertex $u \in U$ represents a user, and edge $u_i \rightarrow u_j \in E$

represents a directed relationship from u_i to u_j. The user graph is also generally represented as an adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$, where N is the number of users. If there is a directed connection from u_i to u_j, the matrix element $A_{ij} = 1$. If there is a weight on each directed edge, indicating the relationship strength, then the matrix element A_{ij} should be the weight w_{ij} of the edge $u_i \rightarrow u_j$.

3.2 Problem Definition

We firstly present general definitions of information diffusion prediction and link prediction problems, and then state the definitions of the target research problems in this thesis.

3.2.1 Information Diffusion Prediction

Diffusion prediction problems are categorized as macro diffusion prediction which estimates future size (popularity) of the diffusion cascade, and micro diffusion prediction which predicts future individuals (participants) of the diffusion cascade. In this thesis, we focus on *micro information diffusion prediction*. Given the previously infected users in a cascade, the goal of it is to predict the next infected user. Formally, given the infection history $\{(u_1, t_1), ..., (u_i, t_i)\}$ of a cascade, the task is to predict user u_{i+1}, who will be infected next.

3.2.2 Link Prediction

Link prediction problems are categorized as temporal link prediction which predicts potential new links on an evolving network, and structural link prediction which infers missing links on a static network. In this thesis, we focus on *structural link prediction*. Given the partially observed structure of a network, the goal of it is to predict the unobserved links. Formally, given a partially observed network $G = (U, E)$, we

represent the set of node-pairs with unknown link status as $E^?$, then the goal of structural link prediction is to infer link status of node-pairs in $E^?$.

3.2.3 Formal Definitions of Research Problems

Research Problem 1

Given only a set of observed (training) diffusion cascades $C = \{c_1, c_2, ..., c_m\}$, the goal is to learn a predictive model that can predict the probability of each infected user $p(u_{i+1}|\{(u_1, t_1), ..., (u_i, t_i)\})$ in future (testing) diffusion cascades.

Research Problem 2

Given a set of observed (training) diffusion cascades $C = \{c_1, c_2, ..., c_m\}$ and a prior user network G, it asks for a predictive model to predict the conditional probability $p(u_{i+1}|\{(u_1, t_1), ..., (u_i, t_i)\})$ in testing cascades.

Research Problem 3

Given a set of observed (training) diffusion cascades $C = \{c_1, c_2, ..., c_m\}$ and a prior user network G, the goal is to learn a predictive model, which is able to predict not only the conditional probability $p(u_{i+1}|\{(u_1, t_1), ..., (u_i, t_i)\})$ for each infected users in testing cascades, but also the linking score (or probability) $p(u_a, u_b)$ for each pair of users in $E^?$.

3.3 Technical Concepts

3.3.1 Social User Representation Learning

Inspired by recent success of representation learning on various research areas, many scholars recently proposed a series of user representation learning methods to understand different user behaviors from different signals in social media. The general goal of these methods is to project users as dense and low-dimensional vectors. The

learned user representations are able to overcome the sparsity problem of traditional one-hot representation or other forms of hand-craft features, and they are expected to capture latent features of behavior data in social media.

Most researchers focused on modeling social linking behavior with user representation learning on social network structure. A series of network representation learning were proposed [107, 127, 56, 23], and summarized by a detailed survey [21]. These methods have been successfully applied to user classification task [35] and link prediction tasks [141]. Some other scholars focused on the user-generated text signal and develop representations learning model to capture the characteristics of content posting behaviors [14]. Besides, some recent studies attempted to learn user representation from multiple signals, which can be widely used in different social network analysis tasks. For example, Li et al. [89] proposed a model that learns user embedding simultaneously from text, social connections and user attributes, and Huo et al. [63] developed a model that jointly learns embedding from user activities and social graph.

Despite the success of user representation learning in social media analysis, a little attention has been paid to user behavior in information diffusion. Bourigault et al. [17] firstly proposed a diffusion model to learn representations of social network users (nodes) in a continuous space, which provides a new perspective for the diffusion prediction problem. Afterwards, Bourigault et al. [18] proposed another representation learning model based on the independent cascade schema. These previous work witnessed the potentiality of representation learning for diffusion modeling.

3.3.2 Recurrent Neural Networks

Recurrent neural network (RNN) [43] was a classical technique for sequence modeling problem. Different from traditional neural network assuming that all inputs (and outputs) are independent, RNNs assume that the output at current step is de-

pendent on the previous computations in the sequence modeling. Therefore, RNNs are able to memorize what has been calculated so far. In theory, RNNs can model sequences with arbitrary length, but in practice, they are limited by the vanishing or exploding gradient problem when the length of sequence increases. To alleviate the problem, some variants of RNNs were proposed, among which long-short term memory (LSTM) [61] and gated recurrent unit (GRU) [31] are the most representative ones. LSTM introduces a memory block, which contains a memory cell and three gates, to compute the hidden state. The three gates, i.e., input, output and forget gates, provide write, read and reset operations for the memory cell, which could access and forget the historical information. Due to high computation cost of gates in LSTM, GRU [31] was proposed to simplify the architecture of the memory block. It removes the memory cell and only two gates are maintained to conduct read and reset operations on hidden states. In some conditions, GRU can achieve competitive or even better performance with lower computation cost.

RNNs have been widely applied in various sequence-based applications, including machine translation [31, 8], session-based recommendation [59, 88, 145], and sequential opinion mining [126, 93]. Recently, a series of RNN-based models [38, 139, 136] were proposed for the diffusion prediction problem. These models have demonstrated the effectiveness of RNNs on modeling diffusion sequences.

3.3.3 Neural Attention Mechanism

To some extent, the proposal of neural attention mechanism is inspired by human vision or reading behavior, in which we pay attention to different regions of an image or words in one sentence. Attention in the deep learning is generally defined as a vector or a matrix of importance weights, indicating how strongly one element, such as one pixel in image or one word in sentence, is correlated with other elements. The neural attention mechanism aims to predict or infer the attention vector and

takes the sum of elements' values weighted by the attention vector as the target representation.

The attention mechanism was initially proposed for inferring correlations between words in source sentences and those in target sentences in machine translation [8]. It was soon developed with various forms [96] and widely applied into different sequence-based tasks, e.g., image captioning [150], sequential recommendation [88] and sentence embedding [92]. In a series of recent works, full attention-based neural networks have proven even more effective than RNN on sequence modeling. The RNN-free attention neural network Transformer [132] was firstly proposed for machine translation task. Based on this work, other attention neural networks [123, 92] were recently proposed for different sequence-based problems.

Part I

Representation Learning Based Models

Chapter 4

Network Regularized Role-based User Representation Learning for Diffusion Prediction

4.1 Chapter Overview

In the literature, most existing diffusion prediction methods focus on modeling interpersonal influence from observed diffusion cascades [118, 57, 112]. Since these models assume that interpersonal diffusion influence of each pair of users is independent, they often suffer from poor generalization ability on capturing hidden diffusion influence. For example, if two users never participate in the same diffusion processes, these models generally estimate the interpersonal influence as 0, indicating that information will never diffuse between them in the future. However, if the two users often interact with their common friends, they will possibly propagate information to each other [149]. This motivates us to model the diffusion by capturing the characteristics of users instead of estimating the diffusion probability attached to the user relationship.

Capturing specific characteristics of users with human designed feature set has been widely studied for many social network analysis tasks, including diffusion prediction [57, 158, 148]. With the recent success of representation learning models,

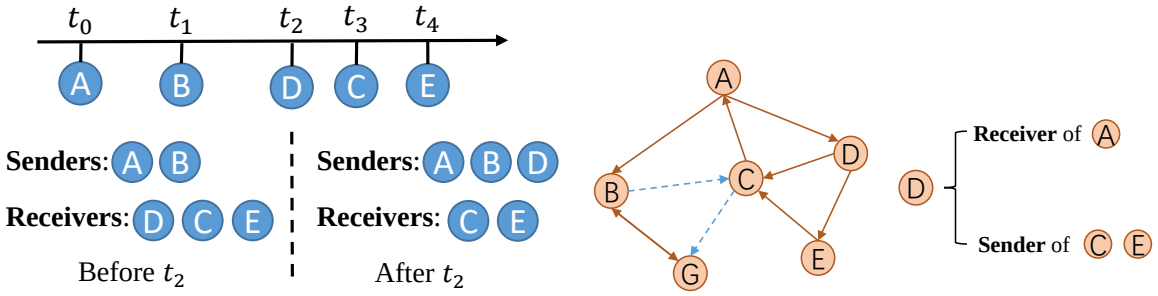
the research focus has moved to the issue of learning user feature representations automatically instead of handcraft feature engineering. The goal of user representation learning is projecting users in a low-dimensional continuous space to understand different user behaviors from different signals in social media. For example, some researchers proposed to learn user representations from network data to model social relationships creation and community formation [127, 35]; some scholars focused on user representation learning from user-generated text data to understand the content posting behavior and infer user preferences on different topics [14]. However, to the best of our knowledge, there is little work on developing a comprehensive user representation learning model for explaining user behaviors in information diffusion. Therefore, in order to bridge this gap, we aim at developing a user representation learning model specific for information diffusion and applying this model to solve the diffusion prediction problem. To achieve this goal, we identify and address the following key issues:

- The first issue is how to simultaneously learn user representations from two separate diffusion-related data sources. Generally, diffusion processes unfold as cascades on the social network. However, observations of diffusion cascades on social media are often actions traces, which only record who participate in diffusion processes at which time. Therefore, we can only obtain sequential form of cascades (Figure 4.1(a)) but cannot observe how and why the information propagates among users. Apart from observed cascades, the network structure, where social connections represent visible diffusion channels, provides another evidence for revealing the diffusion processes [57, 118, 148]. Taking the diffusion cascade in Figure 4.1(a) as an example, it is difficult to judge which user (A, B or D) is the diffusion source of C only from the diffusion observation, but the visible diffusion channel (solid arrow D→C) in the

network structure (Figure 4.1(b)) can help infer that D may be the source. However, the observed social network is often incomplete [134], where some diffusion channels are hidden (blue dashed arrows in Figure 4.1(b)). This will lead to inaccurate inference if the network structure is over-relied. Back to the above-mentioned example, if we over-depend on the given structure (i.e., assigning zero influence on unobserved user pairs), the influence on the hidden link (dashed arrow B→C) would be ignored.

- The second issue is how to capture unique characteristics of information diffusion. One important characteristic is that each user could play as two roles, i.e., Sender and Receiver, in the diffusion process. The two roles can be reflected in both diffusion cascades and network structure. For example, in the cascade of Figure 4.1(a), user D is uninfected before time t_2, so s/he plays as a receiver role and is ready for being infected by previous senders; while after time t_2, D is infected thus s/he becomes a sender and will influence other uninfected users. In the example network (Figure 4.1(b)), D plays as the two roles according to the structural information. For user A, user D acts as a receiver on the directed edge A→D; while for C and E, D is their sender. Therefore, we should carefully consider properties of the two roles for representation learning. Another characteristic of diffusion is that diffusion influence between two users is always asymmetric. For instance, an ordinary user u_n in Twitter always retweets messages from a star user u_s followed by u_n, but this star user u_s rarely retweets messages from u_n. This indicates that diffusion influence from u_s to u_n is much greater than that from u_n to u_s.

To this end, we propose a novel role-based network-regularized user representation learning model for diffusion prediction. The main idea is to project each user with two role-based representations (Receiver and Sender representations), which



(a) Diffusion Cascade (b) User Network

Figure 4.1: Available Diffusion Data Sources and Role-Based Properties

are expected to capture the latent features of role-based properties. To leverage the information from diffusion cascades and user network, we develop a joint learning model that learns the representations by simultaneously optimizing a cascade modeling objective and network regularization objective. The cascade modeling objective aims at maximizing the likelihood of cascades. We formulate the likelihood of cascades by considering the time-weighted aggregated diffusion influence in representation space. We further transfer the objective as an equivalent ranking objective for more efficient optimization. To avoid over-fitting learning on cascades, we propose to apply a network regularization on representation learning. The network regularization is designed to leverage the social network structure information to constrain the learning of user representations. The objective of network regularization is to make the representations reconstruct the structural proximities between users. Since high-order information is considered in the structural proximities, the regularization can effectively avoid the problems of over-dependence on network and potentially capture more hidden diffusion channels in the latent space. An effective and efficient learning algorithm with well-designed sampling strategies is then proposed for this model. The model can be directly applied to diffusion prediction task based on the learned representations. To be concluded, the contributions of this paper are described as follows:

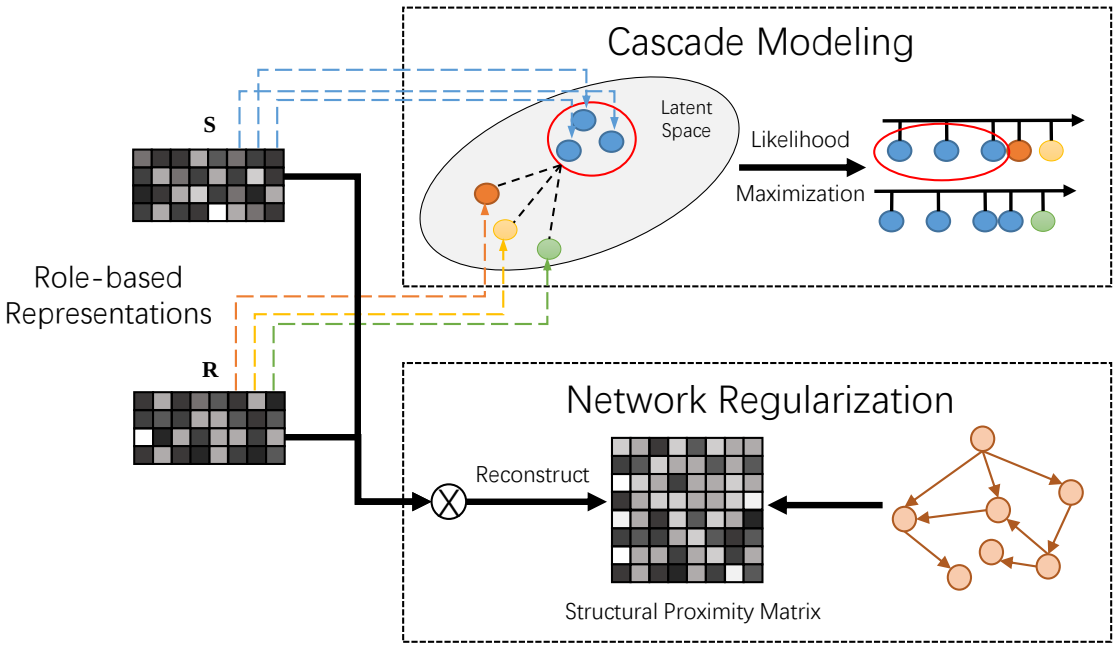


Figure 4.2: Network Regularized Diffusion User Representation Learning Model

- We propose to represent diffusion users as role-based representations to capture latent user-specific characteristics.

- We develop a novel representation learning model to learn role-based user representations from both sequential cascades and network structure. We also implement an effective and efficient learning algorithm for model optimization. The model can be directly used for diffusion prediction problem.

- We evaluate the proposed model on three real-life datasets, including Weibo, Twitter and Memetracker for diffusion simulation and ranking tasks. Compared with the existing state-of-the-art methods, our model achieves better performance significantly and consistently.

4.2 Method

In this work, we propose the Network Regularized Diffusion Representation Learning (NRDR) model. As illustrated in Figure 4.2, the proposed model simultaneously learns role-based representations (i.e., sender and receiver representations) of each user from two diffusion-related signals, i.e., cascades and network. Specifically, the model defines a cascade modeling objective on observed diffusion cascades. It aims at maximizing the likelihood of observed cascades based on role-based representations. This objective is further transferred to an equivalent ranking objective, which is more efficient for learning. The model defines a network regularization objective on network information. It aims at encoding structural information in the representations to avoid over-fitting learning on cascades. A matrix factorization idea is applied to make user representations be able to reconstruct the structural proximities between users. We unified the two objectives in the model and propose an effective and efficient algorithm for model learning. Based on the learned representations, the model can direct predict diffusion process with the proposed cascade modeling framework.

4.2.1 Role-Based User Representation

We propose to project each user u into a latent continuous space $\mathbb{R}^d$ with two role-based representations, i.e., the Sender role representation $\mathbf{s} \in \mathbb{R}^d$ and the Receiver role representation $\mathbf{r} \in \mathbb{R}^d$. The representations aim at capturing two fundamental role-based properties of each user, i.e., *Virality* and *Susceptibility*, whose existence and importance have been proven in previous diffusion research [7, 60]. Virality is the ability of a user to spread information to other users, which takes effects when the user plays as the sender role; while susceptibility is the ability of a user to be infected by possible senders, which is considered when the user is in the receiver role. The $\mathbf{s}$ captures the latent features describing user's virality when s/he is in the

sender role, while the $\mathbf{r}$ captures the features indicating his/her susceptibility when s/he is in the receiver role.

Following previous representation learning models [127, 107], we employ inner product on user representations to measure the pair-wise influence in the latent space. Formally, the diffusion influence from u_i to u_j is defined as $\mathbf{s}_i \cdot \mathbf{r}_j$, where $\cdot$ represents the inner product. It is worth noting that the asymmetry of the diffusion influence is naturally captured by the role-based representations. As for u_i and u_j, we not only consider the diffusion influence from u_i to u_j as $\mathbf{s}_i \cdot \mathbf{r}_j$, but also the influence from u_j to u_i as $\mathbf{s}_j \cdot \mathbf{r}_i$. The computations are different in the latent space, thus the two influence weights are asymmetric.

4.2.2 Representation Learning with Aggregated Influence-Based Cascade Modeling

We propose to learn role-based user representations by modeling observed cascades. Following general cascade modeling framework [118, 112], we set the goal of cascade modeling as maximizing the likelihood of diffusion cascades. Role-based user representations are parameters to be learned. In this work, we follow the general assumptions of information diffusion in previous work [118, 112]: (1) the diffusion state of a user is binary, i.e., either uninfected or infected; (2) the diffusion state can only become infected from uninfected, which indicates no repeated infections occurred in diffusion cascades.

Formally, a cascade describes the following process: *At time step t_i, given the previous infection history $H_{i-1} = \{(u_1, t_1), ..., (u_{i-1}, t_{i-1})\}$, each uninfected user u_j would be infected with the conditional probability $p(u_j|H_{i-1})$; if u_j is successfully infected, then u_j is removed from uninfected set and join the cascade; the same process iteratively occurs until no users are infected in one step.*

It is generally assumed that all infections on a cascade are independent to each

other, then the likelihood of an observed cascade can be represented as the multiplication of the conditional likelihood of each infection $\hat{p}(c) = \prod_{i=1}^{|c|} \hat{p}(u_i|H_{i-1})$. Given a set of observed cascades C, where cascades are independent to each other, then the objective of representation-based cascade modeling is to maximize the following likelihood:

$$\max_{\mathbf{S},\mathbf{R}} \prod_{c \in C} \prod_{i=1}^{|c|} \hat{p}(u_i|H_{i-1}) \tag{4.1}$$

where $\mathbf{S}, \mathbf{R} \in \mathbb{R}^{d \times N}$ are the parameters to be learned. Each column of matrices $\mathbf{S}, \mathbf{R}$ is the role-based representations of each user. $|c|$ is number of infected users in cascade c. $\hat{p}(u_i|H_{i-1})$ is representation-based conditional infection probability estimated by the model.

The conditional probability $p(u_i|H_{i-1})$ is the crucial point in the above learning objective. In this work, we assume that each infection is triggered under the **aggregated influence** of possible senders, which is similar with the assumption of general linear threshold model. At time step t_i, all previously infected users in H_{i-1} acts as information senders while the u_i plays as a receiver. Based on the role representations, we formulate the conditional probability as follow:

$$\hat{p}(u_i|H_{i-1}) = \frac{1}{1 + \exp\left(-\sum_{k=1}^{i-1} w_k(\mathbf{s}_k \cdot \mathbf{r}_i)\right)} \tag{4.2}$$

where $\sum_{w=1}^{i-1} w_k(\mathbf{s}_k \cdot \mathbf{r}_i)$ indicates the weighted-aggregated diffusion influence from previous users on the user u_i. We here employ the sigmoid function to translate the diffusion influence as the infection probability. To guarantee a reasonable probability distribution, we apply normalization to the estimated conditional probability such that the sum of probabilities of all possible next infected users is 1. The normalized

conditional probability is defined as follow:

$$\hat{p}'(u_i|H_{i-1}) = \frac{\sigma\left(\sum_{k=1}^{i-1} w_k(\mathbf{s}_k \cdot \mathbf{r}_i)\right)}{\sum_{u_l \in U_c^{\bar{i}}} \sigma\left(\sum_{k=1}^{i-1} w_k(\mathbf{s}_k \cdot \mathbf{r}_l)\right)} \tag{4.3}$$

where $\sigma(x) = 1/(1+e^{-x})$ represents the sigmoid function, $U_c^{\bar{i}} = U - \{u_1, ..., u_{i-1}\}$ represents the set of all possible next infected users at time t_i.

The weight w_k in the above formula measures the contribution of user u_k in H_{i-1} to the aggregated influence. The influence weight is determined by the time-decay effect, which has proven an important factor in information diffusion [112]. General assumption is that the fresher the infected user is, the more influential s/he is. The computation of w_k is defined as:

$$w_k = \frac{\Delta\left(t_i - t_k\right)}{\sum_{p=1}^{i-1} \Delta\left(t_i - t_p\right)} \tag{4.4}$$

where t_k is the infection time of u_k, t_i is the current time step and $\Delta(\cdot)$ represents a decay effect function of time interval between t_i and t_k, and w_k is normalized by the time intervals of all previous users u_p. We adopt two widely used functions, i.e., Exponential and Rayleigh [112], as $\Delta(\cdot)$. The Exponential function holds monotonic assumption that user importance drops continuously as time passes. The Rayleigh function, however, is non-monotonic, where the importance firstly increases to a peak and then decreases rapidly. The two functions are defined as follow:

$$\Delta_{\text{EXP}}\left(t_i - t_j\right) = e^{-(t_i - t_j)} \qquad \Delta_{\text{RAY}}\left(t_i - t_j\right) = (t_i - t_j)e^{-\frac{\left(t_i - t_j\right)^2}{2}} \tag{4.5}$$

where $t_i > t_j$, if $t_i \leqslant t_j$, then $\Delta(t_i - t_j) = 0$.

Based on the normalized conditional probability $\hat{p}'(u_i|H_{i-1})$ (Equation 4.3), we aim at minimizing the following loss function by applying negative logarithm trans-

formation to Equation 4.1:

$$\mathcal{L}(C) = -\sum_{c\in C}\sum_{i=1}^{|c|}\log\hat{p}'(u_i|H_{i-1}) = -\sum_{c\in C}\sum_{i=1}^{|c|}\left(\log\sigma(\sum_{k=1}^{i-1}w_k(\mathbf{s}_k\cdot\mathbf{r}_i)) - \log\sum_{u_l\in U_c^{\bar{i}}}\sigma(\sum_{k=1}^{i-1}w_k(\mathbf{s}_k\cdot\mathbf{r}_l))\right) \tag{4.6}$$

However, optimizing directly on this loss function suffers a severe computation efficiency problem, since the complexity of term $\log\sum_{u_j\in U_c^{\bar{i}}}\sigma(\cdot)$ approximately reaches as high as $O(N\times|\bar{c}|^2\times|C|)$ in the learning process, where N is the number of all users, $|\bar{c}|$ is the average length of training cascade and $|C|$ is the number of training cascade. It is more reasonable to employ the sampling strategy instead of a complete computation [127].

Inspired by previous ranking-based embedding algorithms [15, 17], which is efficient with negative sampling, we transfer the original objective as a ranking objective. For each infection, maximizing the normalized likelihood $\hat{p}'(u_i|H_{i-1})$ is equivalent to making $\hat{p}'(u_i|H_{i-1}) > \hat{p}'(u_j|H_{i-1})$ and the difference $\hat{p}'(u_i|H_{i-1}) - \hat{p}'(u_j|H_{i-1})$ as much as possible for any uninfected user u_j in cascade ($u_j \neq u_i$). The ranking inequality can be further induced as:

$$\hat{p}'(u_i|H_{i-1}) > \hat{p}'(u_j|H_{i-1}) \Leftrightarrow \frac{\sigma\left(\sum_{k=1}^{i-1}w_k(\mathbf{s}_k\cdot\mathbf{r}_i)\right)}{\sum_{u_l\in U_c^{\bar{i}}}\sigma\left(\sum_{k=1}^{i-1}w_k(\mathbf{s}_k,\mathbf{r}_l)\right)} > \frac{\sigma\left(\sum_{k=1}^{i-1}w_k(\mathbf{s}_k\cdot\mathbf{r}_j)\right)}{\sum_{u_l\in U_c^{\bar{i}}}\sigma\left(\sum_{k=1}^{i-1}w_k(\mathbf{s}_k\cdot\mathbf{r}_l)\right)}$$

$$\Leftrightarrow \sigma(\sum_{k=1}^{i-1}w_k(\mathbf{s}_k\cdot\mathbf{r}_i)) > \sigma(\sum_{k=1}^{i-1}w_k(\mathbf{s}_k\cdot\mathbf{r}_j)) \Leftrightarrow \hat{p}(u_i|H_{i-1}) > \hat{p}(u_j|H_{i-1})$$

Based on above ranking criteria, we employ the widely used margin-based ranking loss [15] and transfer the original problem as the following minimization problem:

$$\min_{\mathbf{S},\mathbf{R}}\mathcal{L}(C) = \sum_{c\in C}\sum_{i=1}^{|c|}\sum_{u_j\in U_c^{\bar{i}},u_j\neq u_i}\max\{1-\left(\sigma(\sum_{k=1}^{i-1}w_k(\mathbf{s}_k\cdot\mathbf{r}_i)) - \sigma(\sum_{k=1}^{i-1}w_k(\mathbf{s}_k\cdot\mathbf{r}_j))\right), 0\} \tag{4.7}$$

where u_j is a sampled user (negative user) based on the truly infected user u_i (positive user) in cascade c. We adopt a pair-wise learning manner. The intuitive goal of this function is to make the likelihood of positive user $\hat{p}(u_i|H_{i-1})$ larger than that of sampled negative user $\hat{p}(u_j|H_{i-1})$, and the likelihood difference should not be smaller than a margin γ. Here we naturally set $\gamma = 1$ for the ground truth of each infection, i.e., $p(u_i|H_{i-1}) - p(u_j|H_{i-1}) = 1$, where the real probability of truly infected user $p(u_i|H_{i-1}) = 1$ and the real probability of uninfected user $p(u_j|H_{i-1}) = 0$. When $\hat{p}(u_i|H_{i-1}) - \hat{p}(u_j|H_{i-1}) < 1$, the function gives a penalty on learning.

4.2.3 Network Regularization on Role-based Representation

As mentioned above, observations from sequential diffusion cascades are limited to capture the diffusion influence adequately or accurately. In addition to the cascades, the network structure is another important factor reflecting users' diffusion influence [60]. The structural information can be employed to complete and correct the learned knowledge from cascades. For instance, given the following situation: (1) there is a pair of linked users u_i and u_j; (2) no diffusion interactions in the training data but some interactions in the test data are recorded, then the diffusion influence learned from cascades may be very low, but the link information indicates that they are still possible to propagate information in the future. Meanwhile, the structural information in network possess consistent role-based characteristics, i.e., (1)intuitively, in a directed relationship $u_i \rightarrow u_j$, u_i plays the information sender and u_j is the receiver; (2) similar to diffusion influence, the pair-wise structural proximity, which measures user similarity based on their structural information on graph, is often asymmetric in directed network [163].

Based on above understandings, we employ the network information as a regularization on the role-based representations to improve the generalization ability of the proposed model. It is expected to preserve the structural proximities in the represen-

tations to avoid over-fitting learning on cascades, which has a similar objective with previous network embedding approaches [107, 127]. A very recent work [110] has proven that most existing network embedding models can be unified into the matrix factorization framework. Considering the flexibility and extensibility, we apply this matrix factorization idea to the network regularization term. Specifically, the network regularization on role-based representations aims at minimizing the following function:

$$\mathcal{R}(G) = \frac{1}{2}||\mathbf{G} - \mathbf{S}^T\mathbf{R}||^2 \tag{4.8}$$

where $\mathbf{G} \in \mathbb{R}^{N\times N}$ is the structural proximity matrix extracted from graph G and $\mathbf{S}$, $\mathbf{R}$ are the matrices of role-based representations. The objective of network regularization is intuitive, i.e., for any pair of users (u_i, u_j), the product of their role-based representations $\mathbf{s}_i \cdot \mathbf{r}_j$ is expected to reconstruct their structural proximity value. The higher proximity value, the structurally closer the two users are.

How to extract the structural proximity matrix $\mathbf{G} \in \mathbb{R}^{N\times N}$ is a crucial point of network regularization. Generally, the network structure is represented as an adjacency matrix $\mathbf{A} = \{A_{ij}\}$, where $A_{ij} = 1$ if there is a directed edge from u_i to u_j. The adjacency matrix can be regarded as the first-order (explicit) information of structural proximity. However, due to the large scale and high dynamics of user network, information of some edges (i.e., first-order information) is often missing in the collected data, the adjacency matrix is very limited to measure the structural proximity. Therefore, many previous studies focused on estimating the structural proximity by combining higher-order (implicit) information. Following the previous work [23], we conduct transformation on $\mathbf{A}$ to extract high-order structural information and employ Positive Point-wise Mutual Information (PPMI) metric to measure the structural proximities between users.

PPMI is originally used in word embedding model [86], and then successfully ap-

plied to network embedding methods [23]. It measures word similarity as $PPMI(w_1, w_2) = \max\{\log \frac{p(w_1,w_2)}{p(w_1)p(w_2)}, 0\}$, where $p(w_1, w_2)$ is co-occurrence probability of words w_1 and w_2 within a t-step context window size in corpus text, and $p(w_1)$ ($p(w_2)$) represents the occurrence probability of w_1 (w_2) in corpus. By casting network users as words and a sufficiently long random walk on network as corpus, the PPMI-based structural proximity is defined as follow:

$$G_{ij} = \max\{\log \frac{M_{ij}}{\sum_k M_{ik} \times \sum_l M_{lj}}, 0\} \tag{4.9}$$

where the matrix $\mathbf{M}$ represents the aggregated structural information of different orders. It is defined as $\mathbf{M} = \frac{\hat{\mathbf{A}}+\hat{\mathbf{A}}^2+\ldots+\hat{\mathbf{A}}^t}{t}$, where $\hat{A}_{ij} = \frac{A_{ij}}{\sum_k A_{ik}}$ represents the exact 1-step probability of reaching to u_j from u_i in random walk (i.e., first-order information) and $\hat{A}^t_{ij}$ is the exact t-step probability (i.e., t^{th}-order information). Therefore, $\hat{M}_{ij}$ is equivalent to the averaged co-occurrence probability of u_i and u_j within t-step window in random walk. $\sum_i M_{ik}$ and $\sum_l M_{lj}$ correspondingly represents the occurrence probability of u_i and u_j. Empirically, we set $t = 2$, since previous work has shown that the second-order information is sufficient to measure the implicit structural proximity [127].

4.2.4 Model Learning

Combining the loss function of cascades modeling $\mathcal{L}(C)$ with the network structure regularization term $\mathcal{R}(G)$, the global objective of the proposed model is as follow:

$$\min_{\mathbf{S},\mathbf{R}} \mathcal{L} = \sum_{c\in C}\sum_{i=1}^{|c|}\sum_{u_j\in U_c^{\bar{i}}, u_j\neq u_i} \max\{1-\left(\sigma(\sum_{k=1}^{i-1} w_k(\mathbf{s}_k\cdot\mathbf{r}_i)) - \sigma(\sum_{k=1}^{i-1} w_k(\mathbf{s}_k\cdot\mathbf{r}_j))\right), 0\} + \frac{\lambda}{2}||\mathbf{G}-\mathbf{S}^T\mathbf{R}||^2 \tag{4.10}$$

where λ denotes the weight parameter of network regularization term. Given the observed cascades set C and the given network G, the objective is to learn representa-

tions $\mathbf{S}, \mathbf{R}$ by minimizing $\mathcal{L}$. The classical stochastic gradient descent (SGD) method is a general solver for both parts of the loss function $\mathcal{L}(C)$ and $\mathcal{R}(G)$. Therefore, a SGD method with designed sampling strategy is applied to solve this optimization problem.

Algorithm 1: Network Regularized Diffusion User Representation Learning

Input : Observed Diffusion Cascades Set C and Network G
Output: User Representation Matrix $\mathbf{S}$ and $\mathbf{R}$

Initialize: $\mathbf{s} \leftarrow$ uniform(-1,1) for each column vector $\mathbf{s}$ in $\mathbf{S}$;
  $\mathbf{r} \leftarrow$ uniform(-1,1) for each column vector $\mathbf{r}$ in $\mathbf{R}$;
  $I \leftarrow 0$;
while $I <$ *Max Iteration* **do**
  for $c \in C$ **do**
    for $u_i \in c$ **do**
      Extract previous infection information $H_{i-1} = \{(u_1, t_1), ..., (u_{i-1}, t_{i-1}\}$;
      Randomly sample a negative user u_j from the set $U_c^{\bar{i}} - \{u_i\}$;
      Calculate $\hat{p}(u_i|H_{i-1}) = \sigma(\sum_{k=1}^{i-1} w_k(\mathbf{s}_k \cdot \mathbf{r}_i))$ and $\hat{p}(u_j|H_{i-1}) = \sigma(\sum_{k=1}^{i-1} w_k(\mathbf{s}_k \cdot \mathbf{r}_j))$;
      if $\hat{p}(u_i|H_{i-1}) - \hat{p}(u_j|H_{i-1}) < 1$ **then**
        Calculate $\frac{\partial l}{\partial \mathbf{s}_k}$, $\frac{\partial l}{\partial \mathbf{r}_i}$, $\frac{\partial l}{\partial \mathbf{r}_j}$ with Equation 4.12, Equation 4.13 and Equation 4.14;
      else
        Calculate $\frac{\partial l}{\partial \mathbf{s}_k}$, $\frac{\partial l}{\partial \mathbf{r}_i}$, $\frac{\partial l}{\partial \mathbf{r}_j}$ with Equation 4.15, Equation 4.16 and Equation 4.17;
      end
      $\mathbf{s}_k \leftarrow \mathbf{s}_k - \alpha \frac{\partial l}{\partial \mathbf{s}_k}$;
      $\mathbf{r}_i \leftarrow \mathbf{r}_i - \alpha \frac{\partial l}{\partial \mathbf{r}_i}$;
      $\mathbf{r}_j \leftarrow \mathbf{r}_j - \alpha \frac{\partial l}{\partial \mathbf{r}_j}$;
    end
  end
  $N \leftarrow N + 1$;
end

The detailed learning algorithm is shown in Algorithm 1. Initially, all users are

projected to the latent space with a uniformly random representation by following the initialization procedure proposed in [49]. Then the learning process will be conducted repeatedly by reversing each infected user in the training cascades. Specifically, for each infected user $u_i \in c$ in a training cascade, we firstly extract the previous infection history H_{i-1}. Then given this positive user u_i and previously infected users in H_{i-1}, we randomly sample a negative user u_j from the user set $U_c^{\bar{i}} - \{u_i\}$. We regard the triplet (H_{i-1}, u_i, u_j) as a training sample, where previously infected users in history H_{i-1} play as senders while positive user u_i or negative user u_j plays as receiver. Based on global objective Equation 4.10, we specifically minimize the following loss for each training sample:

$$l = \max\{1 - \left(\sigma(\sum_{k=1}^{i-1} w_k \mathbf{s}_k \cdot \mathbf{r}_i) - \sigma(\sum_{k=1}^{i-1} w_k \mathbf{s}_k \cdot \mathbf{r}_j) \right), 0\} + \frac{\lambda}{2} \sum_{k=1}^{i-1} \left(||G_{ki} - \mathbf{s}_k \cdot \mathbf{r}_i||^2 + ||G_{kj} - \mathbf{s}_k \cdot \mathbf{r}_j||^2 \right) \tag{4.11}$$

Since $||\mathbf{G} - \mathbf{S}^T\mathbf{R}||^2$ is equivalent to $\sum_{a=1}^{N} \sum_{b=1}^{N} ||\mathbf{G}_{ab} - \mathbf{s}_a \mathbf{r}_b||^2$, when using SGD the loss of each step is $\sum_m \sum_n ||\mathbf{G}_{mn} - \mathbf{s}_m \mathbf{r}_n||^2$ for sampled pairs (u_m, u_n). In our algorithm, for a training triplet (H_{i-1}, u_i, u_j), we have $i-1$ positive pairs (u_k, u_i) and $i-1$ negative pairs (u_k, u_j), where $u_k \in H_{i-1}$. All these sampled pairs should be constrained by network regularization. Therefore, the original form in Equation 4.10 is transformed as above.

The role-based representations of sampled users are updated by taking a gradient step on above loss l with the constant learning rate α. The gradients are calculated by the following equations:

- if $\hat{p}(u_i|H_{i-1}) - \hat{p}(u_j|H_{i-1}) < 1$:

$$\begin{aligned} \frac{\partial l}{\partial \mathbf{s}_k} =& \sigma(\sum_{k=1}^{i-1} w_k \mathbf{s}_k \cdot \mathbf{r}_i)(\sigma(\sum_{k=1}^{i-1} w_k \mathbf{s}_k \cdot \mathbf{r}_i) - 1) w_k \mathbf{r}_i + \sigma(\sum_{k=1}^{i-1} w_k \mathbf{s}_k \cdot \mathbf{r}_j)(1 - \sigma(\sum_{k=1}^{i-1} w_k \mathbf{s}_k \cdot \mathbf{r}_j)) w_k \mathbf{r}_j \\ & - \lambda(G_{ki} - \mathbf{s}_k \cdot \mathbf{r}_i)\mathbf{r_i} - \lambda(G_{kj} - \mathbf{s}_k \cdot \mathbf{r}_j)\mathbf{r}_j \end{aligned} \tag{4.12}$$

$$\frac{\partial l}{\partial \mathbf{r}_i} = \sigma(\sum_{k=1}^{i-1} w_k \mathbf{s}_k \cdot \mathbf{r}_i)(\sigma(\sum_{k=1}^{i-1} w_k \mathbf{s}_k \cdot \mathbf{r}_i) - 1) w_k \mathbf{s}_k - \lambda(G_{ki} - \mathbf{s}_k \cdot \mathbf{r}_i)\mathbf{s}_k \quad (4.13)$$

$$\frac{\partial l}{\partial \mathbf{r}_j} = \sigma(\sum_{k=1}^{i-1} w_k \mathbf{s}_k \cdot \mathbf{r}_j)(1 - \sigma(\sum_{k=1}^{i-1} w_k \mathbf{s}_k \cdot \mathbf{r}_j)) w_k \mathbf{s}_k - \lambda(G_{kj} - \mathbf{s}_k \cdot \mathbf{r}_j)\mathbf{s}_k \quad (4.14)$$

- if $\hat{p}(u_i|H_{i-1}) - \hat{p}(u_j|H_{i-1}) \geqslant 1$:

$$\frac{\partial l}{\partial \mathbf{s}_k} = -\lambda(G_{ki} - \mathbf{s}_k \cdot \mathbf{r}_i)\mathbf{r_i} - \lambda(G_{kj} - \mathbf{s}_k \cdot \mathbf{r}_j)\mathbf{r}_j \quad (4.15)$$

$$\frac{\partial l}{\partial \mathbf{r}_i} = -\lambda(G_{ki} - \mathbf{s}_k \cdot \mathbf{r}_i)\mathbf{s}_k \quad (4.16)$$

$$\frac{\partial l}{\partial \mathbf{r}_j} = -\lambda(G_{kj} - \mathbf{s}_k \cdot \mathbf{r}_j)\mathbf{s}_k \quad (4.17)$$

The learning process will stop until the max number of learning iterations is reached or the learning achieves convergence condition.

4.3 Experiments

4.3.1 Data

In order to test the performance of our method sufficiently, we use 3 datasets from the different online information diffusion channels, including two social networking services, i.e., Weibo and Twitter, and one on-line mainstream news websites network. These three datasets are all representative for the information diffusion study. The detailed statistics of the experimental dataset are shown in Table 4.1.

- **MemeTracker**: This dataset is extracted from the MemeTracker corpus [82]. The original corpus contains articles from mainstream news websites or blogs collected during a specific period. In this dataset, each cascade records the diffusion process of a specific key phrase (or event) and is represented by a sequence of webpage links with a timestamp. To explore the information diffusion

Table 4.1: The Statistics of Experimental Data

Data	Number of Users	Number of Edges	Number of Cascades	Avg. Cascade Length
MemeTracker	1,087	32,932	30,747	8.7
Weibo	8,190	148,752	43,365	21.6
Twitter	13,309	108,657	72,103	9.2

on more diverse channels, we select the articles published on the mainstream news websites in this dataset. We regard each news website as a "user" and create a directed social graph for these websites by using the approach described in [17]. If website A has at least one article, which cites (using a hyperlink as reference) an article published by website B, the approach assumes that there is a directed edge from A to B. We filter out the websites publishing less than 50 articles and extract the corresponding cascades related to these websites.

- **Weibo**: This social media dataset [158] consists of the reposting logs crawled from Sina Weibo, a Chinese Twitter-like microblogging site. In the Weibo network, users are connected by directed following relationships and the information can propagate through users' reposting behavior from their followees. Therefore, the reposting log essentially represents an information diffusion process with temporal orders. We extracted a subset of the original network by filtering out the users who appear less than 30 times in all the reposting logs or are isolated from others. We select the corresponding reposting logs related to these users only.

- **Twitter**: In this dataset [148], users are also linked by the directed following relationships. However, different from the Weibo dataset, this dataset records the diffusion processes of the hashtags in a given social network rather than those of the specific messages (posts or tweets). When a user writes the hashtag

in their tweets, he/she is actually infected by this hashtag. Therefore, we regard the sequences of adopters and timestamps for observed hashtags in the original data as the diffusion processes and extract the experimental dataset in a similar way as the Weibo dataset. As shown in Table 4.1, this dataset has a sparser network structure than the Weibo dataset.

4.3.2 Experiments on Diffusion Simulation

Most diffusion models, especially graph-based models, often apply simulation technique to predict the future diffusion process. In order to testify the simulation performance of the proposed representation learning model, we first propose a diffusion simulation method in representation space and compare the performance with other popular graph-based methods and a state-of-the-art embedding model.

Diffusion Simulation in Representation Space

Based on the learned representations, we propose the corresponding diffusion simulation method for NRDR, which takes the randomness of the diffusion into account. Given a set of source users U_s in a diffusion cascade c, we predict the diffusion as follow:

1. Initialize the infected users set as $U_{in}^c = U_s$ and the uninfected users set as $U_{un}^c = U - U_s$;

2. At each time step t_i, for each user u_i in the uninfected set:

   (a) Calculate the estimated diffusion probability $\hat{p}$ based on Equation 4.2 as follow:

$$\hat{p} = \frac{1}{1 + \exp\left(-\sum_{u_k \in U_{in}^c} w_k(\mathbf{s}_k \cdot \mathbf{r}_i)\right)}$$

(b) Randomly generate a probability p, if $\hat{p} > p$ update $U_{in}^{c} = U_{in}^{c} \cup \{u_i\}$, $U_{un}^{c} = U_{un}^{c} - \{u_i\}$.

3. Repeat step 2 for the next time step, and stop until U_{in}^{c} is no longer updated or the max observation time is reached.

To be consistent with the learning process, the time-decay influence is also considered in the predicting process. The method predicts infected users iteratively. The prediction will stop until no more users in the given network G are infected or the time step has reached the max time.

Baseline Methods

The following methods are introduced as the baselines for diffusion simulation experiments:

- **CTIC** [114]: CTIC refers to Continuous-Time Independent Cascade model, which is a classical explicit graph-based model. The model constraints diffusion channels as links of the prior network structure. It learns diffusion rates and time-decay parameters of these links from observed diffusion cascades. We use the algorithm developed in [114] to learn diffusion parameters for the CTIC model.

- **NetRate** [112]: NetRate is a representative implicit graph-based work that only makes use of sequential data. It infers edges of a implicit diffusion network and estimates transmission rates of each edge that best explain the observed data. It also makes use of the time information in cascades and proposes a series of parametric model with different time-decay distributions over diffusion cascades. Because the NetRate model is generative, once the model is trained it can be used to predict diffusion process based on the inferred graph.

- **EIC** [18]: Embedded Independent Cascade model is a state-of-the-art embedding-based diffusion prediction model. Based on the classical independent cascade schema, EIC embeds users in a latent space to extract more robust diffusion probabilities than those defined by classical graph-based learning approaches. With the similar schema, the trained EIC can be used for diffusion prediction like IC model.

Evaluation Metrics and Settings

The input of the simulation experiment is a seed set of initially infected users S^c (for a testing cascade c). Given this seed set, the model runs iteratively to simulate successive infections based on the learned parameters. The process will stop until no users are newly infected. The output of this process is the predicted set of infected users $\hat{U}^c$.

Since we pay attention to which users will be infected, we evaluate the performance of diffusion simulation on the test set C_t based on the following evaluation metrics. Let the predicted set of infected users be $\hat{U}^c = \{\hat{u}_1, ..., \hat{u}_n\}$ and the ground-truth infected set of cascade c be $U^c = \{u_1, ..., u_m\}$.

- **Precision** measures how many users are correctly predicted in the predicted set:

$$P = \frac{1}{|C_t|} \sum_{c \in C_t} \frac{|\hat{U}^c \cap U^c|}{|\hat{U}^c|}$$

- **Recall** measures how many ground-truth infected users appear in the predicted set:

$$R = \frac{1}{|C_t|} \sum_{c \in C_t} \frac{|\hat{U}^c \cap U^c|}{|U^c|}$$

- **F-1** considers precision and recall together:

$$F1 = \frac{2PR}{P+R}$$

The detailed settings of simulation experiments are mainly referred to the previous work [18]. The size of the seed set in all experiments is 1 to ensure the fairness, i.e., $S^c = \{u_1\}$. In real scenario, we cannot know the time information of future infections. Therefore, it is often a standard setting [18] that simulation process unfolds at unit time per step. The unit time interval of each step is one hour for its better performance on all methods. For each cascade, the process repeats 10000 times and performance is averaged. Furthermore, we conduct 5-fold cross validation and paired t-test to verify the significance of experimental performance. 80% cascades are used for model training and the rest 20% cascades are for testing. The reported results are averaged values of cross validation.

The parameters settings of all compared methods are as follow. As for CTIC, we use the recommended settings in the work [114]. The initial values of time-delay parameter and diffusion parameter are randomly drawn from [0,1] with uniform distribution, and the convergence threshold is 10^{-12}. The settings of NetRate are based on original paper [112], where time window is set as the longest time interval and the time decay pattern is Exponential for its better performance. The training of EIC mainly follows the suggested settings [18]. The dimension size is 30, the learning rate is 10^{-4} and the sampling bias is used. As for the proposed NRDR, the dimension size is same as EIC for fairness, the learning rate is 0.001 and the regularization term weights are 0.001, 0.1 and 0.1 for MemeTracker, Weibo and Twitter, respectively.

Overall Performance

The detailed results of the comparative experiments are reported in Table 4.2.

Table 4.2: Diffusion Simulation Results (Average Performance of 5-Fold Cross Validation)

Data	Methods	Precision	Recall	F-1
MemeTracker	CTIC	0.157	0.415	0.227
	NetRate	0.160	0.396	0.227
	EIC	0.162	0.417	0.233
	NRDR	**0.167***	**0.423***	**0.239***
Weibo	CTIC	**0.126**	0.231	0.163
	NetRate	0.107	0.227	0.145
	EIC	0.122	0.236	0.160
	NRDR	0.125	**0.244***	**0.165***
Twitter	CTIC	0.203	0.408	0.271
	NetRate	0.137	0.389	0.202
	EIC	0.206	**0.411**	0.273
	NRDR	**0.213***	0.410	**0.280***

* indicates better than best competitor at 1% significant level in paired t-test

On the MemeTracker dataset, the proposed NRDR method outperforms other methods when evaluated by all metrics. Meanwhile, NetRate and EIC, which do not use the prior graph information, have relatively better performance than CTIC. The worse results of CTIC can be explained by the artificial links of the created social graph, which may introduce inaccurate structure information. We have confirmed this argument in the following experiments and will give a further explanation. On the other hand, the network regularization in NRDR provides a flexible way of using network information instead of over-depending on it. Therefore, the model is adjustable to the network information with different qualities.

On the Weibo dataset, NRDR performs the best in terms of Recall and F-1. Meanwhile, thanks to the relatively complete network structure of this dataset (the

graph is much denser than the Twitter dataset), the graph-based CTIC obtained a close F-1 performance and even a better precision. On the other hand, the performance of NetRate and EIC drops lower than CTIC. This is because the two methods ignore the network structure, which provides rich information of the diffusion channels in this dataset.

On the Twitter dataset, our method NRDR achieves the best results in terms of precision and F-1. However, CTIC performs not as good as itself on the Weibo dataset and is exceeded by EIC. This performance change may be attributed to the less complete graph information of the Twitter dataset. Since CTIC heavily relies on the prior network structure, its performance is certainly influenced by the incomplete diffusion channels.

Overall, the proposed NRDR outperforms the baselines in terms of most evaluation metrics on the three datasets. Furthermore, our method has a better generalization ability such that it can well handle incompleteness of the diffusion cascades or the prior graph data. Even given the artificial data, it can avoid much errors and achieve satisfactory results.

4.3.3 Experiments on Diffusion Ranking

In this experiment, we focus more on predicting the infection order of users instead of predicting who will be infected. A recent work [17] formulated the diffusion prediction as a ranking problem: given the seed set of infected users, it is expected to rank the uninfected users according to diffusion probabilities. Based on the learned representations, the proposed model can effectively rank the potential users in the latent space. We aim at comparing the ranking performance of NRDR with other state-of-the-art diffusion user representation learning models.

Diffusion Ranking in Representation Space

Given the seed sequence of infected users associated with infection time, we aim at ranking the rest uninfected users based on their representations. Let $H_{in} = \{(u_1, t_1), ...\}$ be the seed infection information, we calculate the infection probability of each potential user u_i as:

$$\hat{p}(u_i|H_{in}) = \sigma(\sum_{(u_k,t_K)\in H_{in}} \frac{\Delta(t_p - t_k)}{\sum_{(u_l,t_l)\in H_{in}} \Delta\left(t_p - t_l\right)} \mathbf{s}_k \cdot \mathbf{r}_i)$$

where we set t_p as 1 hour later than the time of last infection in H_{in}. Then we derive the ranking of uninfected users according to the above formula. The higher the probability is, the higher the ranking is.

Baseline Methods

In this experiment, we consider the following state-of-the-art user representation learning models as baselines. Two of them are network embedding models and the other two are user embedding diffusion models.

- **DeepWalk** [107]: DeepWalk is a popular social network user representation learning model. It learns user representation from network structure. The model employs random walk to sample structure information of network, which is equivalent to diffusion process simulation on the given network structure. Users who frequently co-occur at near positions of same random walks will be closed in latent space. Therefore, distances between user representations of DeepWalk can reflect the diffusion relationship based on the given graph.

- **Node2Vec** [56]: Node2Vec is a more state-of-the-art network embedding model. The framework is similar to DeepWalk, but Node2Vec designs a biased random walk procedure, which is able to capture the diversity of connectivity patterns.

Therefore, the learned representations of Node2Vec are more expressive than those of DeepWalk.

- **CDK** [17]: Content Diffusion Kernel is a state-of-the-art user representation learning model specific for diffusion ranking, which is the most relevant work to ours. This model regards sequential cascades as ranking lists and learns representations with the ranking loss function extracted from the observed cascades. In this model, only the first infected user is treated as diffusion source of cascades and other infected users in the cascades are projected near the source and the distance in the latent space reflects their order of infection. Given the source user, all the other users can be ranked by calculating the distance between their latent representations and the source user's representation.

- **EIC** [18]: EIC is also considered in this experiment.

Evaluation Metrics and Settings

The input of the ranking experiment is also the seed set of initially infected users, and we have known their infection rankings. Given this seed set, we aim at ranking all uninfected users. The output of ranking evaluation is a whole ranking list of all users for a cascade.

We evaluate the performance of diffusion ranking with the classical metric, Mean Average Precision. Let $\hat{I}^c = \{\hat{u}_1, ...\hat{u}_N\}$ be the output ranking sequence of all users for a cascade c (including users in the given seed set). The original sequence c is the natural ground-truth. We calculate the MAP metric for all testing cascades as follow:

$$MAP = \frac{1}{|C_t|} \sum_{c \in C_t} \frac{\sum_{k=1}^{|c|} P_k^c}{|c|}$$

where P_k^c is the precision at top-k of $\hat{I}^c$ for cascade c. The calculation is $P_k^c = \frac{|\{\hat{u}_p\}|}{k}$, where $1 \leqslant p \leqslant k$ and $\hat{u}_p \in c$. $\hat{u}_p$ represents users who are ranked at top-k in $\hat{I}^c$ and also truly infected in ground-truth cascade c.

To ensure the fairness of the comparison, we set the number of dimensions of all methods as 50 in the ranking experiments. Other parameters settings of EIC and NRDR are similar to previous simulation experiments. As for DeepWalk and Node2Vec, window size is 10, walk length is 40 and walks per node is 10. Other parameters of CDK are set for the best performance.

Diffusion Ranking Performance

Since diffusion ranking ignores the interaction process, we conduct experiments with different sizes of seed set to evaluate the ranking performance at different stages of diffusion process. To achieve the ranking results from baselines, we apply the above described ranking method to DeepWalk, Node2Vec and EIC, and employ the original ranking strategy in CDK, i.e., ranking uninfected users by considering their distances to the first user in seed set. Merging the given sequence and the predicted rankings of uninfected users, we can derive the whole ranking list for each diffusion process.

As illustrated in Figure 4.3, the performance is evaluated on the seed size ranging from 1 to 50% of the earlier infected users in cascades, representing different stages of diffusion process. Overall, our model outperforms the other methods on the three datasets in terms of different diffusion stages. On the Weibo dataset, NRDR begins to gain obvious improvements over the other methods when the seed size is larger than 5%. Meanwhile, Node2Vec has better performance than two diffusion embedding models, i.e., CDK and EIC, at most stages. This can be attributed to the relatively complete and accurate network structure of the Weibo dataset, which can provide rich information of diffusion relationships than cascades. When the network is sparse or relatively inaccurate, the performance of network embedding models,

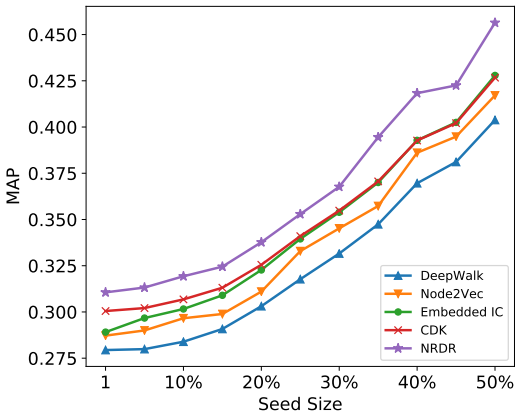
i.e., DeepWalk and Node2Vec, drops down remarkably. Therefore, it is not surprising that DeepWalk achieves the worst performance on the Twitter and MemeTracker datasets. Although NRDR embeds the network structure information as well, it is not very sensitive to the network sparsity or accurateness for its design of network regularization and achieves the best performance on these datasets. Another important finding is that the best competitor CDK can only achieve smallest difference of ranking performance over NRDR at the beginning stage of diffusion process (i.e., seed size is 1). This is because CDK assumes that a newly infected user receives the diffused information only from the first infected user in each cascade. However, this assumption ignores the dynamic of diffusion source since newly infected users often get more diffusion influence from the previous users who are closer to them in cascade or have social connection on the network. The proposed NRDR considers and captures this characteristic well in the representation space, therefore the improvement of NRDR over CDK becomes higher with the increase of the seed size.

4.3.4 Discussions and Analysis

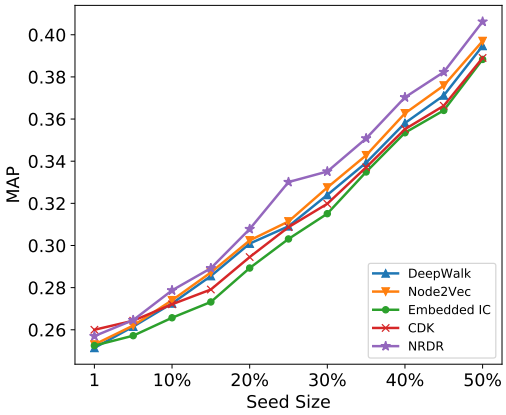
We will further discuss and analyze the parameter sensitivity and the contributions of different components in the proposed model. We introduce the following variants in the ablation studies:

- **NRDR-SR**: It refers to the Single-Role version of our method. In this version, we learn only one representation for each user with ignoring the role difference.
- **NRDR-NN**: It is the No-Net-Structure version of the method. This version ignores the network structure in the learning model. Thus the representations are trained by using the loss function defined as Equation 4.7.

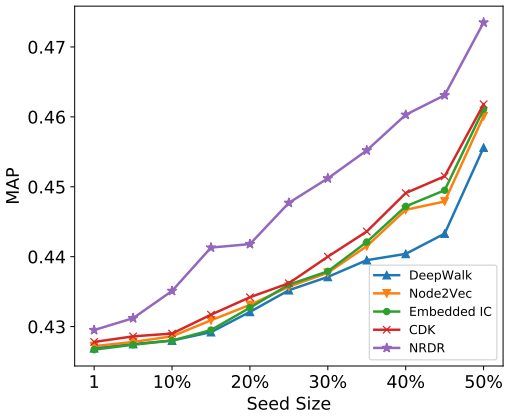
Since the proposed model aims at maximizing the likelihood of cascades, we employ the Negative Cascade Log-Likelihood metric to compare the performance of



(a) MemeTracker Dataset



(b) Weibo Dataset



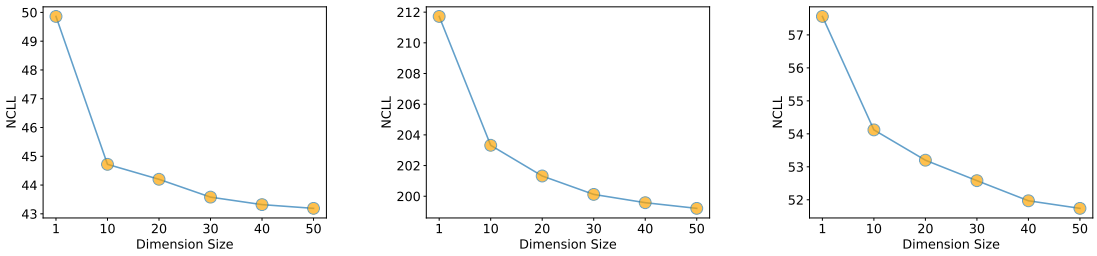
(c) Twitter Dataset

Figure 4.3: Diffusion Ranking with Different Infected Set Size

different parameter settings. The Negative Cascade Log Likelihood is computed as follow:

$$NCLL = -\frac{1}{|C_t|}\sum_{c \in C_t}\sum_{i=1}^{|c|}\log \hat{p}'(u_i|H_{i-1})$$

where we consider the averaged log likelihood of cascades based on the normalized conditional infection probability $\hat{p}'(u_i|H_{i-1})$. The lower NCLL (indicating higher likelihood) the model achieves on testing set, the better prediction ability the model is.



(a) MemeTracker Dataset (b) Weibo Dataset (c) Twitter Dataset

Figure 4.4: The Effect of Dimensionality

The Effect of Dimensionality

The most important setting of the proposed model is the dimensionality of the role-based representations. We compare the performance of the model with different dimension sizes to verify the sensitivity. As shown in Figure 4.4, the model gains more and more improvements with the increase of dimension size. This is not surprising since the representations are more expressive with higher dimension sizes. However, when the dimensionality reaches to a certain level, the improvements will become significantly less. On the other hand, the learning time will also increase when the dimensionality goes up. Therefore, in practice, we should consider the trade-off between performance and efficiency. From the results, dimension sizes of 30 to 50 are relatively better choices for the three datasets.

The Effect of Role-Based Representations

To demonstrate the effectiveness of role-based representations, we compare the proposed NRDR method with its single-role version. As shown in Figure 4.5, NRDR significantly outperforms NRDR-SR on all the datasets. This indicates that the role differentiation is important in modeling diffusion and the role-based representations capture this characteristic more effectively than the single representation for each user. With more expressive latent features of diffusion users, the role-based

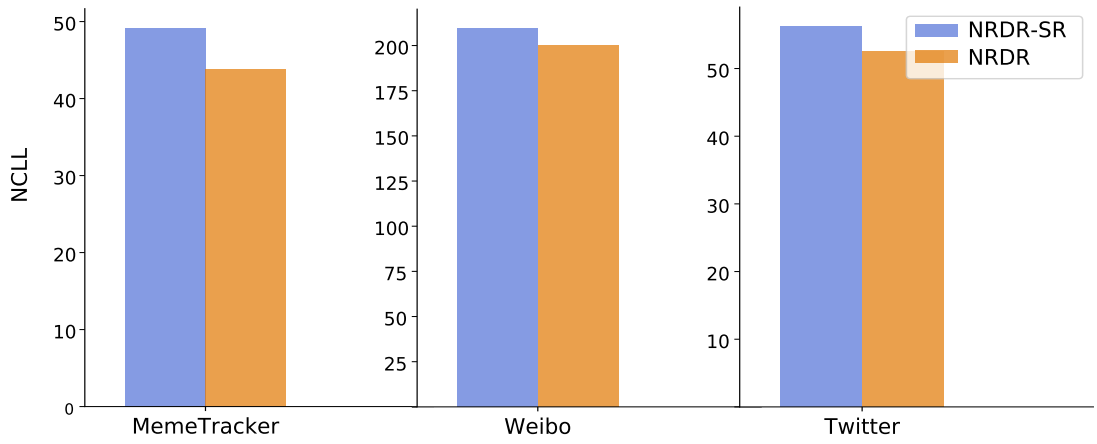


Figure 4.5: Performance Comparison NRDR vs. NRDR-SR

representations achieve significantly better performance than single representation.

The Effect of Network Regularization

In order to verify the importance of network information in the learning model, we compare NRDR with its variant without network regularization, i.e., NRDR-NN, in simulation experiments. As shown in Figure 4.6, NRDR significantly outperforms NRDR-NN on all datasets in the simulation experiments, which demonstrates the importance of introducing the network information into the latent space. With the guidance of structural information, the learned representations are more expressive for diffusion prediction.

To testify the performance improvements are whether from the regularization effect, we further compare the training and testing NCLL of NRDR and NRDR-NN. As shown in Figure 4.7, NRDR-NN suffers the over-fitting problem on Weibo and Twitter datasets, which leads to good convergence on training data but very poor performance on testing data. The network regularization in NRDR effectively alleviates the over-fitting problem and achieves significantly better performance on testing set. The better generalization ability clearly demonstrates the effect of network reg-

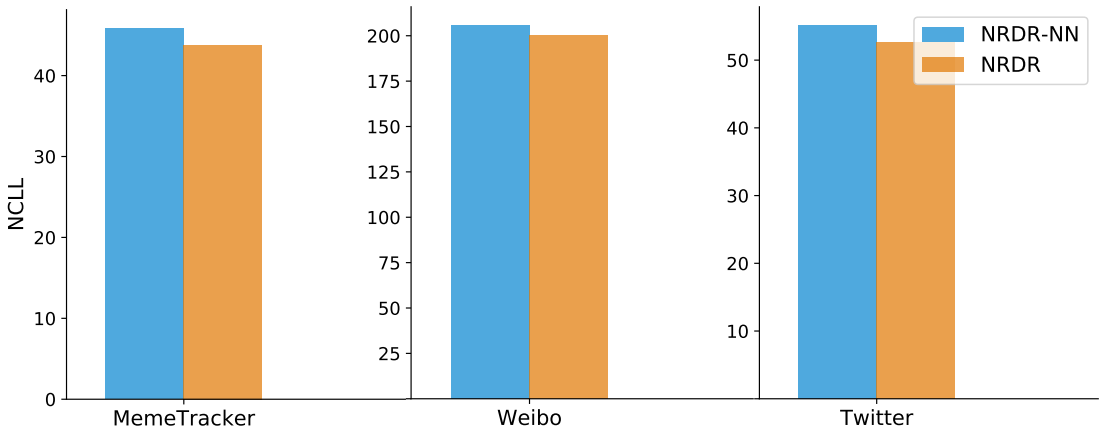
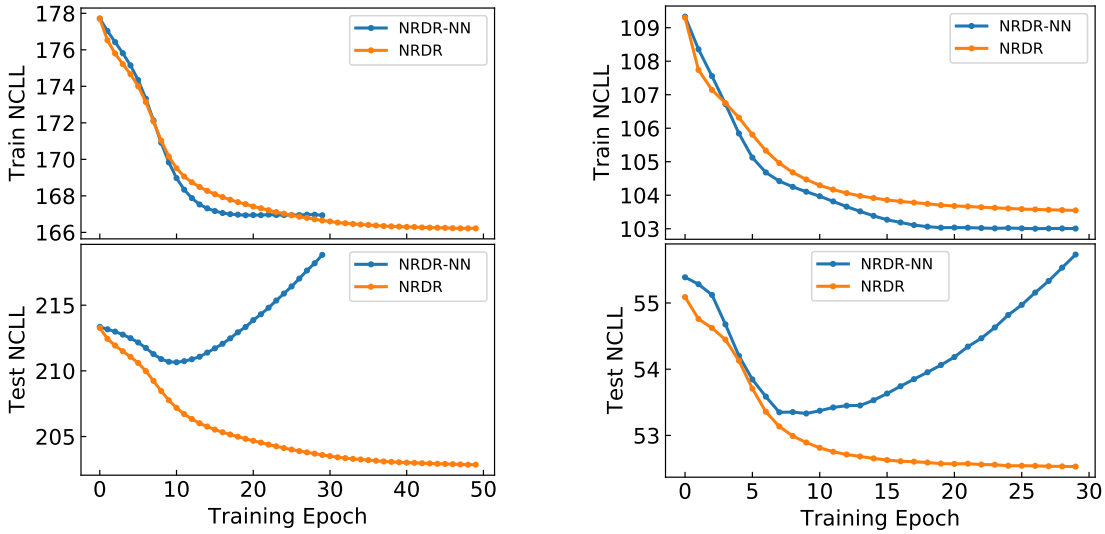


Figure 4.6: Performance Comparison NRDR vs. NRDR-NN

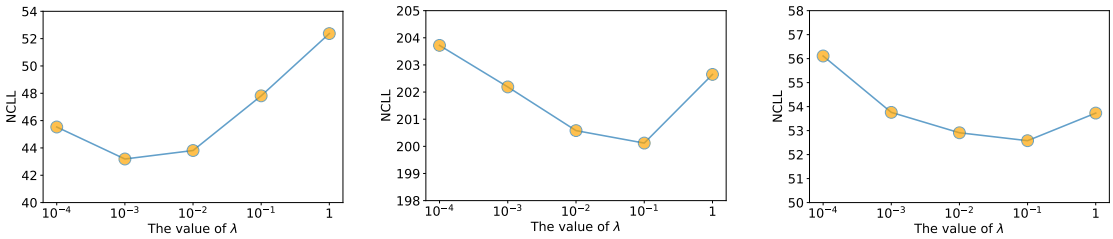
ularization component.

Furthermore, we investigate the performance of the model under different settings of network regularization weight λ. As shown in Figure 4.8, we test the performance with a λ from 10^{-4} to 1, indicating the different degrees that the regularization of the model depends on network information. The proposed NRDR consistently achieves better performance when partially depending on collected network ($\lambda < 1$) than that of complete dependency ($\lambda = 1$). This is mainly because the collected network is often incomplete, where many edges are missing, then partially dependence can avoid over-fitting on incomplete structure. Additionally, the best choices of λ are different on the three datasets. The model prefers a larger λ on Weibo and Twitter datasets, while prefers a relatively smaller λ on MemeTracker dataset. This can be attributed to that the constructed network of MemeTracker is not the actual one since these news sites have no real social relationships like users in social media. In fact, they are all visible to each other and the information flows freely among them. If we utilize an artificial network to regularize the representations, inaccurate structural information could be introduced to harm the result. Therefore, the proposed model favors a relatively lower λ to avoid introducing too much inaccurate structural information.



(a) Weibo Dataset (b) Twitter Dataset

Figure 4.7: The Effect of Network Regularization



(a) MemeTracker Dataset (b) Weibo Dataset (c) Twitter Dataset

Figure 4.8: Performance w.r.t Different Value of λ

Overall, due to the flexibility and adjust-ability of the network regularization, the proposed model has stronger generalization ability on different datasets.

The Effect of Different Time-Decay Functions

As shown in Figure 4.9, we also compare the performance of different time-decay functions. The Exponential function gains significant improvement over the Rayleigh function on the MemeTracker dataset, while the two functions achieve very close performance on Weibo and Twitter datasets. In Weibo and Twitter datasets, there

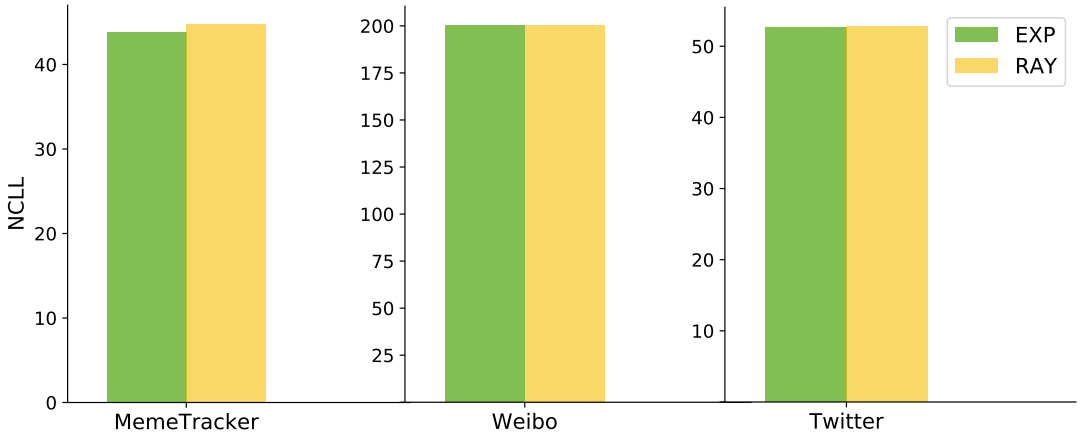


Figure 4.9: Performance w.r.t Different Time Functions

are many cases that infection timestamps of two users are close, but they are far away on the network. In fact, these are random situations where the two users participate the diffusion almost at the same time. The Rayleigh function can handle these situations better. However, in MemeTracker, this kind of cases are relatively rare, thus the Exponential function performs better.

4.4 Chapter Summary

In this chapter, we propose a network-regularized role-based user representation learning model to tackle the information diffusion prediction problem. Instead of representing users on the social graph, we project users into a continuous latent space with role-based representations. We formulate user representation learning by optimizing a cascade modeling objective with a network regularization. We define the representation learning objective on cascades is to maximize the likelihood of cascades. Additionally, we regularize the role-based representations with structural information, which aims at reconstructing pair-wise structural proximities. We develop the prediction method based on the learned representations and compare our method with state-of-the-art methods on three real-world datasets. The experimen-

tal results verify the effectiveness of our model.

Chapter 5

Joint User Representation Learning of Information Diffusion and Network Structure

5.1 Chapter Overview

In previous chapter, we only regard network structure as auxiliary information in our representation learning model. In fact, strong interplay effects exist between information sharing behavior in diffusion cascades and relationship building behavior in network structure, which are intuitive and have already been validated by previous empirical studies [149, 6]. On the one hand, users who show interests to the same messages are more likely to create relationships with each other. Weng et al. [149] conducted an empirical study to investigate how information diffusion flows affected the evolution of the network structure. They verified that the traffic generated by the dynamics of information flow on the network could become an indispensable component for users to connect new friends. Antoniades et al. [6] studied a similar issue on Twitter. They focused on the so-called Tweet-Retweet-Follow behaviors and explained the motivation behind addition of new links under the effect of information sharing. The data analysis showed that a user was more likely to get a new follower if her tweets were retweeted than not retweeted. On the other hand, the

created relationships provide more information channels to further widespread messages, which are also demonstrated by some recent studies [29, 57]. However, current researches on this topic only take analysis to reveal either how diffusion affects network or how network affects diffusion. It is expected to have a comprehensive model that can explore and apply the interplay effects between the two behaviors in real tasks. Few existing user representation learning models pay serious attention to this point. Therefore, there is a strong need to develop a unified representation learning framework that can jointly model the two correlated behaviors with consideration of their mutual effects.

To this end, we explore the interplay effects between information sharing and relationship building behaviors within a unified framework and develop a joint user representation learning model to capture user characteristics based on observed diffusion cascades and social network. The uniqueness of our model is as follows.

1. The user representations are shared by both information diffusion and relationship building behaviors. The shared representations can model not only how users influence each other in information diffusion but also how users build relationship with each other on social network. The correlation of two behaviors is captured in the shared representation space.

2. Similar to our previous work, the representations for each individual user are role-based, according to his/her receiver and sender roles in diffusion and network. A user is characterized by a receiver representation when s/he receives the message from others (or is followed by others) and by a sender representation when s/he spreads out the message (or creating links to others).

To learn the shared representations, we define two learning objectives with respect to two behaviors and combine them in a unified joint learning framework. In particular, the diffusion objective is to maximize the likelihood of generating observed

diffusion cascades while the relationship objective aims to maximize the probability of generating network links to social neighbors. Moreover, we design an efficient sampling-based algorithm to optimize the proposed joint model. As for diffusion objective, instead of directly computing the likelihood of all users in a diffusion cascade, the algorithm randomly samples users to maximize the likelihood of the sampled users in the cascade. As for relationship objective, instead of directly computing the neighbor probability conditional on all users, the algorithm randomly draws negative users to discriminate the real target neighbors from negative neighbors. The user representations are updated by optimizing the two objectives alternatively with the Stochastic Gradient Descent Method (SGD). Due to the generative property of the proposed model, the learned representations can be directly applied to diffusion prediction and link prediction.

We summarize the main contributions of this work as follows.

- We explore the correlation between information sharing and relationship building behaviors with behavior-shared user representations. To differentiate user's receiver and sender roles in both behaviors, we propose to learn two role-based representations for each user in the shared space.

- We develop a joint learning model to learn user representations simultaneously from both diffusion cascade and social network information. We also implement an efficient algorithm for model optimization.

- The proposed user representation learning model is applicable to any correlated behavior modeling and is able to tackle other similar tasks. Moreover, it is easy to be extended to cope with multiple behaviors to provide a more comprehensive view of diffusion users.

We evaluate the proposed model on two important social media tasks, i.e., diffu-

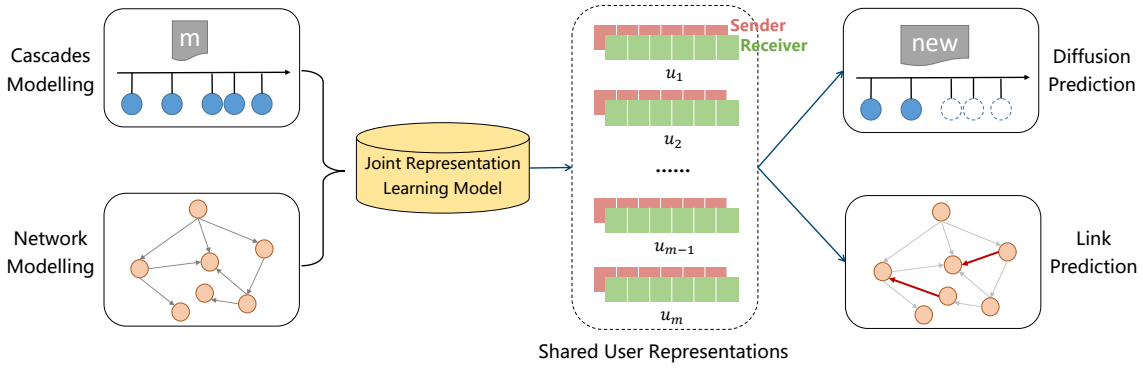


Figure 5.1: Joint User Representation Learning Framework.

sion prediction and link prediction. The experiments are conducted on two popular datasets, which are collected from two real-life social media websites, i.e., Twitter and Weibo. Compared with state-of-the-art methods, the proposed model shows significant superiority on two prediction tasks. The experimental results verify that modeling the correlation of two behaviors is beneficial to both tasks. By learning from both behaviors, the proposed model also shows better robustness when the training data of one behavior is artificially reduced.

5.2 Method

To simultaneously model the user characteristics from diffusion processes and network structure, we propose the joint **D**iffusion-**N**etwork user **R**epresentation **L**earning model (**DNRL**). The idea is illustrated as Figure 5.1. The shared representations are learned to embed the two signals comprehensively in the continuous latent space. Based on these user representations, we are able to predict the future diffusion process and the hidden links.

5.2.1 User Representations

User representation learning aims at projecting each user $u \in U$ into a low-dimensional latent space $\mathbb{R}^n$, where n is the dimension of user vectors. In order to differentiate the characteristics of different user roles in the diffusion process and network structure, we project each user u into two role-based representations. Specifically, when user u is not infected in a specific diffusion process or s/he is the child in a directed edge, u plays as a Receiver and should be projected as the Receiver representation $\mathbf{r}_u$ in the latent space; u transfers his/her role to a Sender when s/he is already infected in a specific diffusion process or s/he is the parent in a directed edge, thus u should be represented as the Sender representation $\mathbf{s}_u$ under this circumstance.

5.2.2 Representation Learning with Independent Influence-Based Cascade Modeling

Similar to previous chapter, the representation-based cascade modeling objective is to maximize the likelihood of observed diffusion cascades. In this work, we follow the general assumptions of information diffusion in previous work [112]: (1) the diffusion state of a user is binary, i.e., either uninfected or infected; (2) the diffusion state can only become infected from uninfected, which indicates no repeated infections occurred; (3) each already-diffused user activates uninfected users independently

Given a set of observed cascades $C = \{c\}$, we assume all cascades are independently generated, thus the overall likelihood can be formulated as multiplication of each cascade likelihood $\ell(C) = \prod_{c \in C} \ell(c)$. For each observed cascade $c = \{(u_1, t_1), ..., (u_n, t_n)\}$, we call $\{u_1, ..., u_n\}$ as "infected" users while $U - \{u_1, ..., u_n\}$ as "survival" users in cascade c, where n users are infected and $|U| - n$ users survive from this cascade, where $|U|$ is the set of all users. Different from previous work, we not only consider likelihood of each infection but also that of each survival (non-infection) in a diffusion cascade. Regarding all infections and survivals as independent events,

then the overall representation learning objective on observed cascade set is defined as follow:

$$\max_{\mathbf{S},\mathbf{R}} \ell(C) = \prod_{c \in C} \ell(c) = \prod_{c \in C} \ell^I(c) \times \ell^S(c), \tag{5.1}$$

where $\ell^I(c)$ represents the likelihood of all infection events in c and $\ell^S(c)$ indicates the likelihood of all survival events. $\mathbf{S}$ and $\mathbf{R}$ denote the matrices of the role-based representations to be learned. We further transfer the original problem to the following minimization problem by employing negative log loss function:

$$\min_{\mathbf{S},\mathbf{R}} \mathcal{L}_C = -\sum_{c \in C} \log \ell(c) = -\sum_{c \in C} (\log \ell^I(c) + \log \ell^S(c)) \tag{5.2}$$

In previous chapter, we have explored the aggregated influence assumption in cascade modeling. In this chapter, we will adopt an independent-cascade-like assumption that each infection can be triggered by **independent influence** of previously infected users. This means that previously infected users have chances to infect future users independently. Based on this assumption, we define likelihoods $\ell^I(c)$ and $\ell^S(c)$ as follows.

Likelihood of Survivals

Assume the max observation of cascade c is T^c, then if user u_m survives from c ($t_m > T^c$), it means that already-infected users $\{u_1, ..., u_n\}$ fail to infect u_m during $[0, T^c]$. We define the likelihood that user u_m survives from cascade c as the multiplication of all pair-wise survivals:

$$\ell^S(u_m, c) = \prod_{\{j | t_j < T^c\}} f_S(u_j, t_j, u_m, T^c) \tag{5.3}$$

where $f_S(\cdot)$ is a function of calculating the likelihood that u_m is survived from u_j who is infected in c at time t_j. Then the likelihood of all survival events in c is:

$$\ell^S(c) = \prod_{\{m | t_m > T^c\}} \prod_{\{j | t_j < T^c\}} f_S(u_j, t_j, u_m, T^c) \tag{5.4}$$

Likelihood of Infections

If user u_m is infected in c at t_m ($t_m < T^c$), then it should be only infected by one of previous infected users before t_m. For example, if u_m is infected by u_i, then it should survive from all the other previously infected users whose infection time is earlier than t_m. The the likelihood of this example can be represented as: $f_I(u_i, t_i, u_m, t_m) \times \prod_{\{j|t_j<t_m, j\neq i\}} f_S(u_j, t_j, u_m, t_m)$ where $f_I(\cdot)$ is a function of calculating the likelihood that u_m is infected by u_i at time t_m. By summing over the likelihood of all such possible infection situations for user u_m, we can derive the likelihood of the infection of u_m in c as:

$$\begin{aligned}
\ell^I(u_m, c) &= \sum_{\{i|t_i<t_m\}} f_I(u_i, t_i, u_m, t_m) \times \prod_{\{j|t_j<t_m, j\neq i\}} f_S(u_j, t_j, u_m, t_m) \\
&= \prod_{\{j|t_j<t_m\}} f_S(u_j, t_j, u_m, t_m) \times \sum_{\{i|t_i<t_m\}} \frac{f_I(u_i, t_i, u_m, t_m)}{f_S(u_i, t_i, u_m, t_m)} \\
&= \prod_{\{i|t_i<t_m\}} f_S(u_i, t_i, u_m, t_m) \times \sum_{\{i|t_i<t_m\}} \frac{f_I(u_i, t_i, u_m, t_m)}{f_S(u_i, t_i, u_m, t_m)}
\end{aligned} \tag{5.5}$$

Then the likelihood of all infection events over c can be defined as:

$$\ell^I(c) = \prod_{\{m|t_m<T^c\}} \prod_{\{i|t_i<t_m\}} f_S(u_i, t_i, u_m, t_m) \sum_{\{i|t_i<t_m\}} \frac{f_I(u_i, t_i, u_m, t_m)}{f_S(u_i, t_i, u_m, t_m)} \tag{5.6}$$

Representation-Based Infection and Survival Functions

The key point in above likelihoods is how to define infection and survival functions parameterized by user representations. We ground the definitions on the classical survival analysis theory [81, 80], which is widely used in previous work [112]. In survival analysis, the two functions aim at capturing diffusion time-decay effects. The infection function $f_I(u_i, t_i, u_j, t_j)$ is a probability density function of the infection time interval $t_j - t_i$. The cumulative distribution function $F_I(u_i, t_i, u_j, t_j) =$

$\int_{t_i}^{t_j} f_I(u_i, t_i, u_j, t_j)$ represents the probability that u_j is infected by u_i within period $(t_i, t_j]$. Therefore, the survival function, which represents the probability that u_i is not infected by u_j not later than t_j, can be defined as $f_S(u_i, t_i, u_j, t_j) = 1 - F_I(u_i, t_i, u_j, t_j)$.

The Exponential time decay distribution has shown robust performance in diffusion modeling [112], therefore we define the two functions with the Exponential form as follow:

$$f_I(u_i, t_i, u_j, t_j) = \alpha_{ij} e^{-\alpha_{ij}(t_j - t_i)} \tag{5.7}$$

$$f_S(u_i, t_i, u_j, t_j) = e^{-\alpha_{ij}(t_j - t_i)} \tag{5.8}$$

where α_{ij} represents the pairwise diffusion influence in the representations space. In the latent space, the distance between user representations can be regarded as a natural index of diffusion influence. The closer the representations are, the higher diffusion influence is. Meanwhile, the value of diffusion influence should be not smaller than 0 and be normalized in a certain interval to avoid great variance. Based on these properties, we apply sigmoid function $\sigma(\cdot)$ to define the representation-based diffusion influence as follow:

$$\alpha_{ij} = \lambda \sigma(-\frac{\beta}{2}||\mathbf{s}_i - \mathbf{r}_j||^2) = \frac{\lambda}{1 + \exp\left(\frac{\beta}{2}||\mathbf{s}_i - \mathbf{r}_j||^2\right)} \tag{5.9}$$

where β is the parameter to control the scale of representation distance, λ is a scale parameter of influence and $\sigma(\cdot)$ represents sigmoid function. In this chapter, we also utilize role-based user representations, we can capture directed diffusion influence. α_{ij} represents the influence from u_i to u_j, thus it is computed with the sender representation u_i and receiver representation of u_j.

5.2.3 Representation Learning with Network Structure

The network structure provides possible channels for information diffusion. In order to explore the correlation between diffusion and network, the proposed model should

embed the information of the observed channels in user representations. The information of structural channels related to a specific user can be naturally translated to the neighborhood information of the user. The learning objective from network thus comes down to preserve neighborhood information of each user according to the observed network structure. Additionally, the neighborhood information of each user is role-based. Specifically, when user u plays as sender, s/he cares about the users who can receive message from him/her, thus the neighbors of u are his/her children under this circumstance; while when user u becomes receiver, s/he cares about users who can send message to him/her. In this situation, the neighborhood of u consists of his/her parents.

The idea of preserving neighborhood information is similar to that of the Skip-Gram word embedding model [100], where word representations can predict frequent surrounding words (context). In our scenario, role-based neighbors of a user can be treated as frequent "context". By using maximum likelihood (ML) principle, the representation learning objective is to maximize the probability that each user generates its role-based neighbors for observed edges. Specifically, when u_i plays as sender, the probability that u_i generates its child u_j, who plays as receivers, is defined as follow:

$$p_1(u_j|u_i) = \frac{\exp\left({\mathbf{s}_i}^T \cdot \mathbf{r}_j\right)}{\sum_{n=1}^{|U|} \exp({\mathbf{s}_i}^T \cdot \mathbf{r}_n)} \tag{5.10}$$

Similarly, when u_i play as receiver, the probability that u_i generates its parent u_h, who is sender the for u_i, is defined as follow:

$$p_2(u_h|u_i) = \frac{\exp\left({\mathbf{r}_i}^T \cdot \mathbf{s}_h\right)}{\sum_{n=1}^{|U|} \exp({\mathbf{r}_i}^T \cdot \mathbf{s}_n)} \tag{5.11}$$

By introducing negative-log transformation, we can formulate the learning objec-

tive as minimizing the following loss functions:

$$\mathcal{L}_{N_1} = -\sum_{(i,j)\in E} \log p_1(u_j|u_i) = -\sum_{(i,j)\in E} \log \frac{\exp\left({\mathbf{s}_i}^T \cdot \mathbf{r}_j\right)}{\sum_{n=1}^{|U|} \exp({\mathbf{s}_i}^T \cdot \mathbf{r}_n)} \tag{5.12}$$

$$\mathcal{L}_{N_2} = -\sum_{(h,i)\in E} \log p_2(u_h|u_i) = -\sum_{(h,i)\in E} \log \frac{\exp\left({\mathbf{r}_i}^T \cdot \mathbf{s}_h\right)}{\sum_{n=1}^{|U|} \exp({\mathbf{r}_i}^T \cdot \mathbf{s}_n)} \tag{5.13}$$

The direct computation of above normalized probabilities is very expensive. With the aim of user representation learning, we do not need a full probabilistic model. The model is instead trained using a binary classification objective to discriminate the real target neighbors from noise. For the sake of improving computation efficiency, we apply an approximation method called Negative Sampling (NEG) [100], then the above loss function can be formulated as follow:

$$\mathcal{L}_{N_1} = -\sum_{(i,j)\in E} \{\log \sigma({\mathbf{s}_i}^T \mathbf{r}_j) + \sum_{k}^{K} \mathbb{E}_{u_k^{\text{out}} \sim P_{\text{neg}}^{\text{out}}(u)} [\log \sigma(-{\mathbf{s}_i}^T \mathbf{r}_k^{\text{out}})]\} \tag{5.14}$$

$$\mathcal{L}_{N_2} = -\sum_{(h,i)\in E} \{\log \sigma({\mathbf{r}_i}^T \mathbf{s}_h) + \sum_{k}^{K} \mathbb{E}_{u_k^{\text{in}} \sim P_{\text{neg}}^{\text{in}}(u)} [\log \sigma(-{\mathbf{r}_i}^T \mathbf{s}_k^{\text{in}})]\} \tag{5.15}$$

where $\sigma(\cdot)$ denotes the sigmoid function, K is the number of negative samples. $P_{\text{neg}}^{\text{out}}(u)$ and $P_{\text{neg}}^{\text{in}}(u)$ are the noise distributions of users according to different roles. Following Mikolov et. al. [100], we use the 3/4 power distribution form for $P_{\text{neg}}^{\text{out}}(u)$ and $P_{\text{neg}}^{\text{in}}(u)$. When u_i is sender, for the edge (i, j), K negative samples are randomly drawn from $P_{\text{neg}}^{\text{out}}(u) \propto (d_u^{\text{out}})^{3/4}$, where d_u^{out} is the out-degree of u in G. When u_i is receiver, for the edge (h, i), K negative samples are randomly drawn from $P_{\text{neg}}^{\text{in}}(u) \propto (d_u^{\text{in}})^{3/4}$, where d_u^{in} is the in-degree of u in G.

The above functions indicate two different views of representation learning from network. $\mathcal{L}_{N_1}$ considers out-neighbors from a sender-centric view, while $\mathcal{L}_{N_2}$ focuses on in-neighbors from a receiver-centric view. We incorporate these two views in one

learning objective to capture structural properties comprehensively. By considering the out-neighborhood of source user and the in-neighborhood of target users on each directed link, we combine $\mathcal{L}_{N_1}$ and $\mathcal{L}_{N_2}$ as the following loss function:

$$\begin{aligned}\mathcal{L}_N = &-\sum_{(i,j)\in E}\{\log\sigma({\mathbf{s}_i}^T\cdot\mathbf{r}_j)+\sum_{k}^{K}\mathbb{E}_{u_k^{\text{out}}\sim P_{\text{neg}}^{\text{out}}(u)}[\log\sigma(-{\mathbf{s}_i}^T\cdot\mathbf{r}_k^{\text{out}})]\\ &+\sum_{k}^{K}\mathbb{E}_{u_k^{\text{in}}\sim P_{\text{neg}}^{\text{in}}(u)}[\log\sigma(-{\mathbf{r}_j}^T\cdot\mathbf{s}_k^{\text{in}})]\}\end{aligned} \tag{5.16}$$

5.2.4 Joint Model and Optimization

We propose to learn user representations simultaneously from diffusion cascades and network structure to embed the characteristics of both behaviors in the same latent space. To achieve this goal, we incorporate the above two learning objectives in a unified model as follow:

$$\min_{\mathbf{S},\mathbf{R}}\mathcal{L}=\min_{\mathbf{S},\mathbf{R}}(\mathcal{L}_C+\mathcal{L}_N) \tag{5.17}$$

In order to solve the above problem efficiently, we propose a joint learning algorithm (Algorithm 2). Inspired by the idea of multi-task representation learning in deep neural networks, we apply a similar learning strategy to optimize the two objectives iteratively. This learning framework uses gradient descent based methods to optimize the two subproblems. The rest parts of this section will discuss the details of training processes on the two subproblems.

Optimization on Diffusion Cascades

Recall the Equation 5.2, $\mathcal{L}_C$ is the sum of negative-log likelihood $-log\ell(c)$ for different cascades. The equation of $-log\ell(c)$ can be expanded based on Equation 5.6 and

Algorithm 2: Joint User Representation Learning

Input : Training diffusion cascades set $C = \{c\}$, training network $G = (U, E)$;

Output: User representations matrices $\mathbf{S}$, $\mathbf{R}$

Initialize: $\mathbf{s_u}, \mathbf{r_u} \leftarrow \text{uniform}(-1, 1)$ for each user u ;

repeat

  One Iteration Learning on $\mathcal{L}_C$:

  Update parameters according to $\mathcal{L}_C$, with learning rate e_1;

  One Iteration Learning on $\mathcal{L}_N$:

  Update parameters according to $\mathcal{L}_N$, with learning rate e_2;

until *Convergence*;

Equation 5.4 as follow:

$$
\begin{aligned}
&-log\ell(c) = -(\log \ell^I(c) + \log \ell^S(c)) \\
&= \sum_{\{m|t_m<T^c\}} \Bigg(\underbrace{-\sum_{\{i|t_i<t_m\}} \log f_S(u_i,t_i,u_m,t_m) - \log \sum_{\{i|t_i<t_m\}} \frac{f_I(u_i,t_i,u_m,t_m)}{f_S(u_i,t_i,u_m,t_m)}}_{\mathcal{L}_{C_1}\text{(Infection Negative-Log Likelihood)}} \Bigg) \\
&+ \sum_{\{m|t_m>T^c\}} \Bigg(\underbrace{-\sum_{\{j|t_j<T^c\}} \log f_S(u_j,t_j,u_m,t_m)}_{\mathcal{L}_{C_2}\text{(Survival Negative-Log Likelihood)}} \Bigg)
\end{aligned}
\tag{5.18}
$$

where $\mathcal{L}_{C_1}$ denotes the negative-log loss of individual infected situation and $\mathcal{L}_{C_2}$ represents the loss of individual survival situation. By replacing the survival and infection functions with Equation 5.7 and Equation 5.8, $\mathcal{L}_{C_1}$ and $\mathcal{L}_{C_1}$ can be transformed to the representation based loss functions as follow:

$$
\mathcal{L}_{C_1} = \sum_{\{i|t_i<t_m\}} \lambda\sigma(\frac{\beta||\mathbf{s}_i - \mathbf{r}_m||^2}{-2})(t_m - t_i) - \log \sum_{\{i|t_i<t_m\}} \lambda\sigma(\frac{\beta||\mathbf{s}_i - \mathbf{r}_m||^2}{-2}) \tag{5.19}
$$

$$
\mathcal{L}_{C_2} = \sum_{\{j|t_j<T^c\}} \lambda\sigma(-\frac{\beta||\mathbf{s}_j - \mathbf{r}_m||^2}{2})(T^c - t_j) \tag{5.20}
$$

Algorithm 3: One Iteration Learning on $\mathcal{L}_C$

for *i in* $0 \sim ts_c$ **do**
  Randomly sample a cascade $c \in C$;
  Randomly sample a user $u_m \in U$;
  if u_m *is infected in* c **then**
    for *each* $u_i \in \{u_i | t_i < t_m\}$ **do**
      $\mathbf{s}_i \leftarrow \mathbf{s}_i - e_1 \frac{\partial \mathcal{L}_{C_1}}{\partial \mathbf{s}_i}$;
      $\mathbf{r}_m \leftarrow \mathbf{r}_m - e_1 \frac{\partial \mathcal{L}_{C_1}}{\partial \mathbf{r}_m}$;
    end
  end
  if u_m *is not infected in* c **then**
    for *each* $u_j \in \{u_j | t_j < T^c\}$ **do**
      $\mathbf{s}_j \leftarrow \mathbf{s}_j - e_1 \frac{\partial \mathcal{L}_{C_2}}{\partial \mathbf{s}_j}$;
      $\mathbf{r}_m \leftarrow \mathbf{r}_m - e_1 \frac{\partial \mathcal{L}_{C_2}}{\partial \mathbf{r}_m}$;
    end
  end
end

In this way, given a specific cascade c and a target user u_m, we can regard such pair (c, u_m) as a training sample for $\mathcal{L}_C$. Whether u_m being infected in c or not decides the different forms of loss functions. The amount of such training samples is very large, thus we propose to employ Stochastic Gradient Descent (SGD) with mini-batch method (Algorithm 3), which is very efficient for this subproblem. In one training iteration, ts_c pairs of (c, u_m) are randomly sampled. If u_m is infected in c, the gradients are calculated based on $\mathcal{L}_{C_1}$. For each $u_i \in \{u_i | t_i < t_m\}$, we update their sender representations with the following gradient:

$$\begin{aligned} \frac{\partial \mathcal{L}_{C_1}}{\partial \mathbf{s}_i} &= \frac{\lambda(t_m - t_i)\partial \sigma(-\frac{\beta||\mathbf{s}_i - \mathbf{r}_m||^2}{2})}{\partial \mathbf{s}_i} - \frac{\partial \sigma(-\frac{\beta||\mathbf{s}_i - \mathbf{r}_m||^2}{2})}{\sum_p \sigma(-\frac{\beta||\mathbf{s}_p - \mathbf{r}_m||^2}{2}) \partial \mathbf{s}_i} \\ &= -\beta \alpha_{im}(1 - \frac{\alpha_{im}}{\lambda}) \left((t_m - t_i) - \frac{1}{\sum_p \alpha_{pm}} \right) (\mathbf{s}_i - \mathbf{r}_m) \end{aligned} \tag{5.21}$$

where $\sigma(\cdot)' = \sigma(\cdot)(1 - \sigma(\cdot))$ and α is calculated as Equation 5.9. Given each above

Algorithm 4: One Iteration Learning on $\mathcal{L}_N$

for i *in* $0 \sim ts_n$ **do**
  Randomly sample $(i, j) \in E$;
  Randomly draw K negative users $\{u_k^{\text{out}}\}$ and K negative users $\{u_k^{\text{in}}\}$ from $P_{\text{neg}}^{\text{out}}(u_i)$ and $P_{\text{neg}}^{\text{in}}(u_j)$ respectively;
  $\mathbf{s}_i \leftarrow \mathbf{s}_i - e_2 \frac{\partial \mathcal{L}_N}{\partial \mathbf{s}_i}$;
  $\mathbf{r}_j \leftarrow \mathbf{r}_j - e_2 \frac{\partial \mathcal{L}_N}{\partial \mathbf{r}_j}$;
  for *each* u_k^{out} *and* u_k^{in} **do**
    $\mathbf{r}_k^{\text{out}} \leftarrow \mathbf{r}_k^{\text{out}} - e_2 \frac{\partial \mathcal{L}_N}{\partial \mathbf{r}_k^{\text{out}}}$;
    $\mathbf{s}_k^{\text{in}} \leftarrow \mathbf{s}_k^{\text{in}} - e_2 \frac{\partial \mathcal{L}_N}{\partial \mathbf{s}_k^{\text{in}}}$;
  end
end

possible sender u_i, we update the receiver representation of u_m with the following gradient:

$$\frac{\partial \mathcal{L}_{C_1}}{\partial \mathbf{r}_m} = \beta \alpha_{im} (1 - \frac{\alpha_{im}}{\lambda}) \left((t_m - t_i) - \frac{1}{\sum_p \alpha_{pm}} \right) (\mathbf{s}_i - \mathbf{r}_m) \tag{5.22}$$

If u_m is not infected in c, then we update representations with gradients calculated on $\mathcal{L}_{C_2}$. For each $u_j \in \{u_j | t_j^c < T^c\}$, we update their sender representation with following gradient:

$$\frac{\partial \mathcal{L}_{C_2}}{\partial \mathbf{s}_j} = -\beta \alpha_{jm} (1 - \frac{\alpha_{jm}}{\lambda}) (T^c - t_m) (\mathbf{s}_i - \mathbf{r}_m) \tag{5.23}$$

Similarly, given above each u_j we update $\mathbf{r}_m$ for the sampled u_m with the gradient:

$$\frac{\partial \mathcal{L}_{C_2}}{\partial \mathbf{r}_m} = \beta \alpha_{jm} (1 - \frac{\alpha_{jm}}{\lambda}) (T^c - t_m) (\mathbf{s}_i - \mathbf{r}_m) \tag{5.24}$$

Optimization on Network Structure

As for learning on $\mathcal{L}_N$, we also use SGD with mini-batch to be consistent with the learning on $\mathcal{L}_C$. We optimize the subproblem on network from per-edge view. With designed negative sampling strategy, we propose the Algorithm 4 to update user

representations from network data. ts_n represents the batch size of one learning iteration on $\mathcal{L}_N$. For each sampled edge (i, j) in batch, we correspondingly draw negative samples from two noise distributions. For sender u_i, negative receivers are sampled from $P^{\text{out}}_{\text{neg}}(u_i)$; and for receiver u_j, negative senders are drawn from $P^{\text{in}}_{\text{neg}}(u_i)$. Then the related representations are updated with the gradients calculated on $\mathcal{L}_N$. For the source user u_i of sampled edge (i, j), we update the sender representation of u_i with following gradient:

$$\begin{aligned}\frac{\partial \mathcal{L}_N}{\partial \mathbf{s}_i} &= \frac{\partial \log \sigma(\mathbf{s}_i{}^T \cdot \mathbf{r}_j)}{\partial \mathbf{s}_i} + \sum_k^K \frac{\partial \log \sigma(-\mathbf{s}_i{}^T \cdot \mathbf{r}_k^{\text{out}})}{\partial \mathbf{s}_i} \\ &= (1 - \sigma(\mathbf{s}_i{}^T \cdot \mathbf{r}_j))\mathbf{r}_j - \sum_k^K (1 - \sigma(-\mathbf{s}_i{}^T \cdot \mathbf{r}_k^{\text{out}}))\mathbf{r}_k^{\text{out}}\end{aligned} \tag{5.25}$$

For the target user u_j of the sampled edge, the receiver representation $\mathbf{r}_j$ is updated similarly with the following gradient:

$$\frac{\partial \mathcal{L}_N}{\partial \mathbf{r}_j} = (1 - \sigma(\mathbf{s}_i{}^T \cdot \mathbf{r}_j))\mathbf{s}_i - \sum_k^K (1 - \sigma(-\mathbf{r}_j{}^T \cdot \mathbf{s}_k^{\text{in}}))\mathbf{s}_k^{\text{in}} \tag{5.26}$$

Additionally, we need to update the representations of negative users. For each negative out user, we update their receiver representations with the following gradient:

$$\frac{\partial \mathcal{L}_N}{\partial \mathbf{r}_k^{\text{out}}} = \frac{\partial \log \sigma(-\mathbf{s}_i{}^T \cdot \mathbf{r}_k^{\text{out}})}{\partial \mathbf{r}_k^{\text{out}}} = (1 - \sigma(-\mathbf{s}_i{}^T \cdot \mathbf{r}_k^{\text{out}}))\mathbf{s}_i \tag{5.27}$$

while for each negative user, the gradient for updating their sender representations is as follow:

$$\frac{\partial \mathcal{L}_N}{\partial \mathbf{s}_k^{\text{in}}} = \frac{\partial \log \sigma(-\mathbf{r}_j{}^T \cdot \mathbf{s}_k^{\text{in}})}{\mathbf{s}_k^{\text{in}}} = (1 - \sigma(-\mathbf{r}_j{}^T \cdot \mathbf{s}_k^{\text{in}}))\mathbf{r}_j \tag{5.28}$$

Table 5.1: The Statistics of Experimental Data

Data	#Users	#Links	#Cascades	Avg. Cascade Length
Weibo	8,190	148,752	43,365	21.6
Twitter	13,309	108,657	72,103	9.2

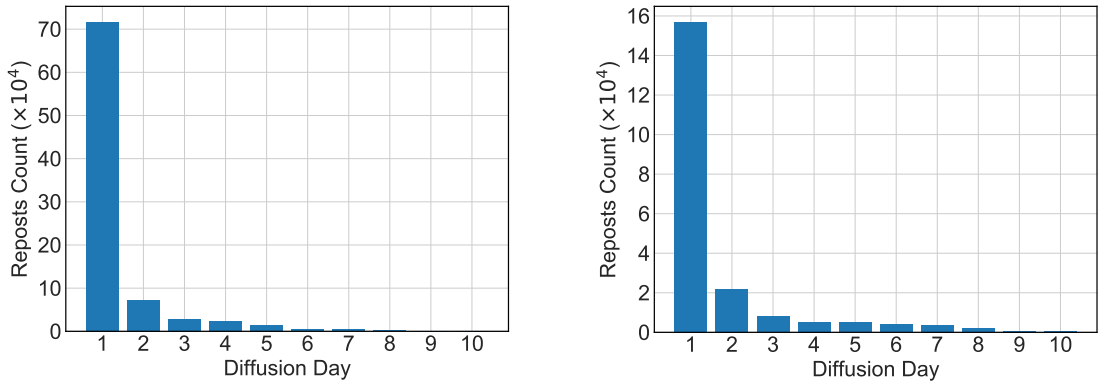
5.3 Experiments

5.3.1 Datasets

In order to verify the performance of the proposed method, we use representative datasets from two popular social networking services, i.e., Weibo and Twitter. Both datasets contain a sampled network from the whole platform and record the diffusion logs among users of the sampled network during a certain period.

Following previous work [112], we assume the max observation time is unified for all diffusion cascades, i.e., $T^c = T$. Existing work often sets the max observation time as one day [76]. To verify whether this setting is suitable on our experimental datasets, we made analysis on the diffusion delay time delay (from source user). The Figure 5.2 illustrates the diffusion counts distribution over time delay (per-day). It is found that the majority of diffusions can be observed within one day on both datasets, thus T is 1 day.

We split the data into training and testing sets by selecting a posting time (the time stamp of source user). The cascades posted before this timestamp are treated as training (observed diffusion) data (80%), while the rest as testing (future diffusion) set (20%). The final experimental datasets are summarized in Table 5.1. The network of Weibo dataset is relatively denser than that of Twitter dataset. Correspondingly, the cascades of Weibo are relatively longer than those of Twitter.



(a) Weibo Dataset (b) Twitter Dataset

Figure 5.2: Diffusion Counts Distribution Over Time

5.3.2 Diffusion Prediction

Compared Methods

We consider the following state-of-the-art diffusion models as experimental baselines.

- **CTIC** [114]: Continuous-Time Independent Cascade (CTIC) model is a classical graph-based model. The model assumes the information can only be diffused through links of the network and considers the impact of time delay with exponential distribution. We use the method proposed in [114] to train the CTIC model. Delays and diffusion influence parameters of each edge are learned conjointly by an EM algorithm.

- **EIC** [18]: Embedded Independent Cascade (EIC) is a state-of-the-art representation learning model for diffusion prediction. Instead of inferring edge weights on graph, EIC projects each user into a latent continuous space only based on diffusion cascades and represents pair-wise diffusion probabilities with latent space distances. Therefore, EIC does not make use of network information.

- **DRL**: DRL is a variant of the proposed model, referring Diffusion Representation Learning. This variant learns user representations only based on the

learning objective of diffusion cascades (Equation 5.2). Therefore, this baseline does not take network information into account.

- **NE-EIC**: Since all existing representation learning method on diffusion only embeds user from diffusion cascades, we developed a combined baseline, Network Embedding with EIC (NE-EIC), to leverage the network information in representation learning. Concatenation has been widely used to combine embeddings learned from different signals as a comprehensive representation [127]. Following the popular idea, we concatenate the user representations learned by EIC with trained network embeddings. Here, we employ DeepWalk [107] to train the network embeddings, since DeepWalk shows better ability of structure preservation in the following link prediction experiments.

Evaluation Metrics and Settings

In this experiment, we evaluate the performance of diffusion prediction from two views, i.e., individual view and global view.

In the individual view, we focus on the probability of each individual user being infected in the diffusion processes. Given the current infected users set of a cascade and infection time, we aim at predicting the next infected user. Since there is no ground truth of individual infection probabilities for all users, we only consider the probabilities of truly infected users observed in testing set. The higher the predicted probabilities of these infections are, the better performance the diffusion model has. The log-likelihood is widely used as a standard in most previous work. Therefore, we employ the following metric, Infection Log-Likelihood (ILL), to evaluate the individual predictability:

$$\textbf{ILL} = \frac{1}{|C| \cdot |c|} \sum_{c \in C} \sum_{\{i | t_i^c < T\}} \log \hat{P}\left(u_i^c | E_i^c\right)$$

where $\hat{P}\left(u_i^c|E_i^c\right)$ is the predicted probability that u_i^c is infected by current infected set E_i^c at time t_i^c, $|c|$ represents the number of infections in c and C is the set of testing cascades. The proposed model can calculate $\hat{P}\left(u_i^c|E_i^c\right)$ based on Equation 5.5. To guarantee the fairness, the predicted probability of truly infected user is normalized by the probabilities of all potential users, such that all models have a consistent probability scale with the constraint $\sum \hat{P}\left(u_p|E_i^c\right) = 1$. The higher the ILL is, the better the predictability is.

In the global view, we pay more attention to the generation probability of whole diffusion cascades. The proposed models and baselines are all generative, thus the likelihood that a testing cascade is generated can be directly obtained based on the learned parameters. Following [18], we evaluate all testing cascades with the metric averaged Cascade Log-Likelihood (**CLL**) as follow:

$$\mathbf{CLL} = \frac{\sum_{c \in C} \log \hat{P}\left(c\right)}{|C|}$$

where $\hat{P}\left(c\right)$ is the estimated probability that cascade c is generated. The higher CLL is, the more accurate the learned model is.

The parameters settings in this experiment are reported as follows: the dimension size is set as 50 for EIC, NE-EIC, DRL and DNRL. As for other settings of DRL and DNRL, the λ is 2, the β is 1; the learning rates of cascades and network are set as 0.015, 0.015 for the Weibo Dataset, and 0.025, 0.015 for the Twitter Dataset, respectively. The CTIC is trained based on the default settings introduced in [114]. The training of EIC also follows the suggested settings in [18]. In the NE-EIC method, we keep the same settings as EIC to learn embeddings from cascades and the settings for network embeddings follows default settings in [107].

Overall Performance

The overall results are shown in Table 5.2. The proposed model significantly outperforms other methods in terms of both metrics. Even though there is no network information, the degenerated variant of the proposed model DRL can still perform much better than other methods. This indicates that the representation learning schema of the proposed model better models the diffusion influence from the diffusion cascades. With jointly embedding the incomplete network information, the model also achieves a significant improvement. Besides, both CTIC and NE-EIC have higher result than EIC. This demonstrates the importance of network structure for diffusion prediction. However, the two methods still have disadvantages. CTIC holds a heavy dependency on the graph structure, which will lead to an inaccurate inference of diffusion weights on edges. The embedding concatenation of NE-EIC is too simple to capture the effect of structure, therefore it receives few improvements over EIC. Our model succeeds to overcome the above disadvantages. By projecting network and diffusion cascades information in a shared representation space, the proposed model does not need any graph assumptions. Meanwhile, with joint representation learning framework, the information of the two behaviors is captured sufficiently and effectively, leading to higher expressiveness of the learned representations.

In the following sections, we will give a further discussion on the robustness of the proposed model in different conditions.

Performance w.r.t. Observed Diffusion Cascades Number

In social media, diffusion interactions among users are highly frequent. The number of observed diffusion cascades is very limited. In this experiment, we aim at investigating the robustness of the proposed model when the number of training cascades

Table 5.2: Diffusion Prediction Results

	Weibo		Twitter	
Methods	ILL	CLL	ILL	CLL
CTIC	-12.24	-252.15	-11.97	-105.07
EIC	-12.99	-269.12	-13.01	-111.83
NE-EIC	-12.87	-268.73	-12.98	-111.19
DRL	-10.35	-201.59	-11.34	-94.72
DNRL	**-9.11**	**-189.11**	**-9.50**	**-82.10**

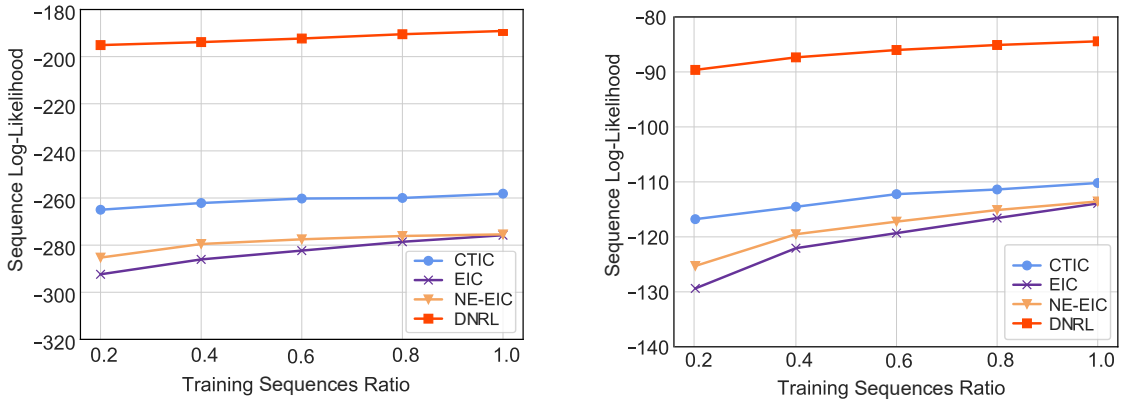
reduces.

The results are shown in Figure 5.3. In total, the proposed model still outperforms baselines with different training ratios. We find that methods with using network information (i.e., DNRL, CTIC and NE-EIC) are more robust than EIC that only utilizes cascades data. When the observed cascades are not sufficient to well learn diffusion influence between users, the network information can become a strong evidence to infer the diffusion influence. Among methods using network information, DNRL shows much stronger robustness with keeping higher performance when the number of training cascades reduces.

Performance w.r.t. Network Completeness

In order to further investigate the impact of network structure on diffusion prediction performance, we conduct experiments with different completeness of given network. The results are shown in Figure 5.4, where the training network ratio represents the rate of edges of original network used for training and performance is measured by CLL. Since EIC and DRL do not leverage network information, their performance does not change in the experiments.

The different degrees of network completeness do affect the performance of the proposed model and CTIC. On the other hand, NE-EIC obtains few benefits from in-

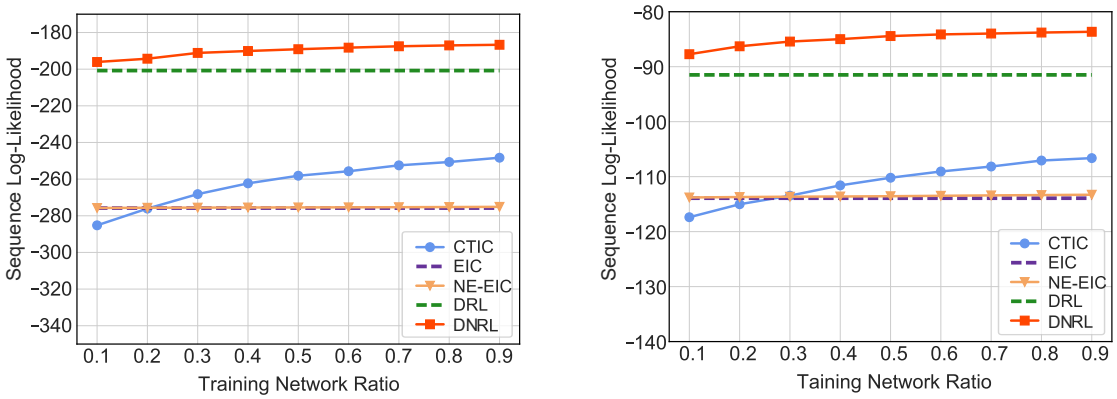


(a) Results on Weibo Dataset

(b) Results on Twitter Dataset

Figure 5.3: Diffusion Prediction Performance with Different Training Diffusion Cascades Ratio

tegrating network information. Even though more network information is embedded, the improvement is still extremely slight. This further proves that embedding concatenation of NE-EIC is an ineffective way to integrate network information. Another important finding is that the proposed DNRL is more robust than CTIC on incomplete network structure. The performance of CTIC decreases more dramatically than the proposed model when the network completeness decreases. In particular, when the completeness ratio is smaller than 20%, CTIC even performs worse than EIC, which does not use the network information. This can be attributed to the different ways of using network information in CTIC and DNRL. CTIC is highly dependent on the discrete network structure with the assumption that infections happen along observed edges; while DNRL releases the heavy dependence by embedding discrete graph formation into a continuous space. Without the hard constraints, it can effectively reduce the risk of depending on incomplete network information.



(a) Results on Weibo Dataset (b) Results on Twitter Dataset

Figure 5.4: Diffusion Prediction Performance with Different Network Completenesses

5.3.3 Link Prediction

Compared Methods

In this experiment, we aim at evaluating the missing link prediction ability of learned representations. Recent network embedding models are effective on preserving structural information and their abilities of inferring missing links have been demonstrated. Therefore, we consider the following state-of-the-art network embeddings models as baselines:

- **DeepWalk** [107]: DeepWalk is a popular user representation learning model. It learns the user representation from the network structure. The model employs random walk to sample the structure information of network. Users who frequently co-occur at near positions of same random walks will be close in latent space.

- **LINE** [127]: LINE aims at preserving the first-order and second-order proximity between users separately. It applies node-pair sampling instead of random walk to extract structural information, thus it is more efficient on large-scale network than DeepWalk. LINE1 represents the method that preserves first-

order proximity and LINE2 represents that for second-order proximity.

- **NRL**: NRL is another variant of the proposed model, referring Network-only Representation Learning. This variant learns user representations singly based on the learning objective defined with network structure (Equation 5.16). The diffusion cascades are not leveraged for representation learning.

- **NE-EIC**: To introduce diffusion cascades information into network embedding methods, NE-EIC is compared in the following experiment. Similar to previous settings, we apply DeepWalk as the network embedding method.

Evaluation Metric and Settings

The link prediction quality is measured by a standard metric, the area under the receiver operating characteristic curve (AUC). The calculation of AUC requires prediction scores (probabilities) of each node-pair in test set. In the proposed model and the variant NRL, we calculate the predicted probability of a node-pair (i, j) as follow:

$$p(i, j) = \frac{1}{1 + \exp(-\mathbf{s}_i^T \mathbf{r}_j)} \tag{5.29}$$

The probability calculation of other baselines is similar to above formula, the only difference is the inner product. In DeepWalk and LINE1, only single embedding of each user can be used to compute the pair-wise inner product; while in LINE2, we calculate the inner product between the node embedding and the context embedding; in NE-EIC, calculation is conducted between concatenated sender embedding and receiver embedding.

In each experiment, we randomly remove edges from original network with a fixed proportion ranging from 10% to 50%, while ensuring that the residual network after edge removals is connected. Meanwhile, we select the same number of non-edged

node pairs as the negative samples in each experiment. The testing set consists of the removed edges and negative samples. The rest of the network is treated as training network. In this way, we can regard the link prediction task as a binary classification problem, i.e., classifying whether a node-pair is linked or not.

Since the proposed model is able to simultaneously predict diffusion and missing links, the parameter settings are same as in the diffusion prediction experiments. The variant of NRL has similar settings with the network structure learning part. For LINE, the learning rate of the starting value is 0.025. The number of negative samples is set as 5 and the total number of samples is 1 billion. For DeepWalk, the window size is 10, the walk length is 40 and walks per vertex is set as 40. For NE-EIC, we follow the settings of EIC in diffusion prediction experiment and the same settings of DeepWalk.

Link Prediction Results

The results of link prediction on both datasets are shown in Table 5.3. The results of the proposed DNRL and its variant NRL are consistently superior to the state-of-the-art network embedding methods on both datasets. Unlike the compared network embeddings, we aim at preserving not only the structural information of their out-neighbors (children) but also that of in-neighbors (parents). With this more comprehensive learning schema, we can better capture the properties of different roles (sender and receiver) that users play in the graph. Additionally, the better performance of DNRL than its variant NRL implies that the diffusion information can indeed help the network embedding to better predict the missing links. We will further investigate the effect of diffusion information in the following experiment. Moreover, concatenating network embedding with diffusion representations even drags down the performance of DeepWalk. This represents that embedding concatenation is also an ineffective way to introduce diffusion information for link

Table 5.3: Link Prediction Results

(a) AUC Scores on Weibo Data

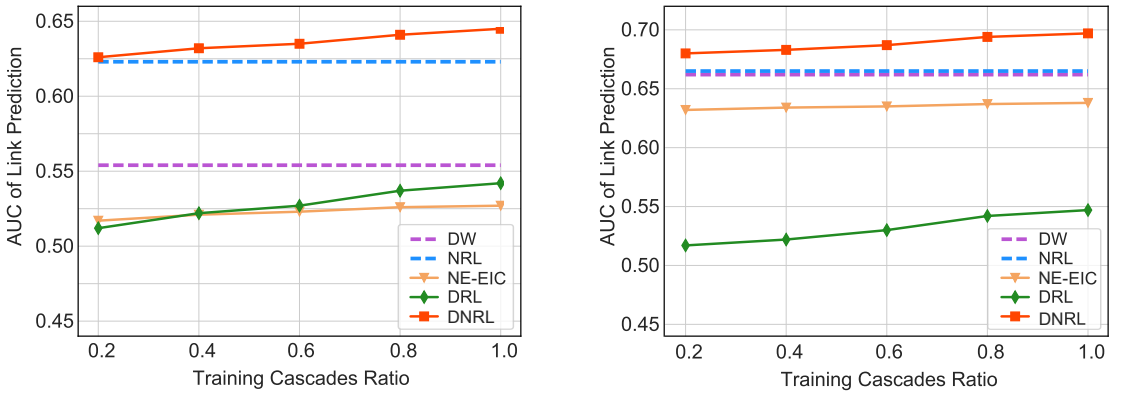
	Fraction of training edges				
Methods	50%	60%	70%	80%	90%
DeepWalk	0.554	0.557	0.560	0.563	0.564
LINE1	0.516	0.517	0.521	0.521	0.525
LINE2	0.521	0.538	0.542	0.548	0.552
NE-EIC	0.527	0.529	0.530	0.531	0.532
NRL	0.623	0.625	0.626	0.627	0.629
DNRL	**0.645**	**0.648**	**0.652**	**0.654**	**0.655**

(b) AUC Scores on Twitter Data

	Fraction of training edges				
Methods	50%	60%	70%	80%	90%
DeepWalk	0.662	0.665	0.669	0.672	0.676
LINE1	0.660	0.664	0.667	0.668	0.666
LINE2	0.526	0.520	0.552	0.538	0.543
NE-EIC	0.632	0.634	0.636	0.636	0.639
NRL	0.665	0.667	0.670	0.673	0.674
DNRL	**0.680**	**0.682**	**0.689**	**0.694**	**0.698**

prediction. Different from simple concatenation, the proposed DNRL captures the correlated effects between network and diffusion by jointly learning the shared representations, thus shows a significant improvement.

On the other hand, the results of DeepWalk and LINE1 gain a large increase on the Twitter dataset compared with their performance on Weibo dataset as shown in Table 5.4(b). This is mainly because user relationships of Twitter dataset are reciprocal. Since we need to guarantee the residual network is connected when creating testing set, removed edges in testing set have the corresponding edge with



(a) Results on Weibo Dataset (50% network) (b) Results on Twitter Dataset (50% network)

Figure 5.5: Link Prediction Performance with Different Training Diffusion Cascades Ratio

opposite direction in the training set. The network embedding methods that learn single representation for each user can benefit a lot from this, because the predicted probability of a testing edge is same with that of its opposite-directed training edge if each user has only single embedding (Equation 5.29). But this cannot help to improve the performance of LINE2, which learns two embeddings for each user as the proposed model. In spite of this situation, the proposed DNRL can even obtain a further improvement over best competitors, which should be the contribution of embedded diffusion cascades information.

Effect of Diffusion on Link Prediction

We further investigate the effect of introducing diffusion information when preserving network structural information. We design the link prediction experiment with providing different numbers of observed cascades to the proposed model. In this experiment, we also evaluate the variant DRL to prove whether the diffusion cascades information can help improve the ability of inferring missing links.

The results are illustrated in Figure 5.5. On both datasets, NE-EIC continuously performs worse than DeepWalk and gains few improvements though the number of

Table 5.4: Effect of Role-Based Representation

(a) Diffusion Prediction Results

	Weibo		Twitter	
Methods	ILL	CLL	ILL	CLL
Single	-12.01	-224.73	-11.84	-98.08
Role-Based	**-9.11**	**-189.11**	**-9.50**	**-82.10**

(b) Link Prediction Results

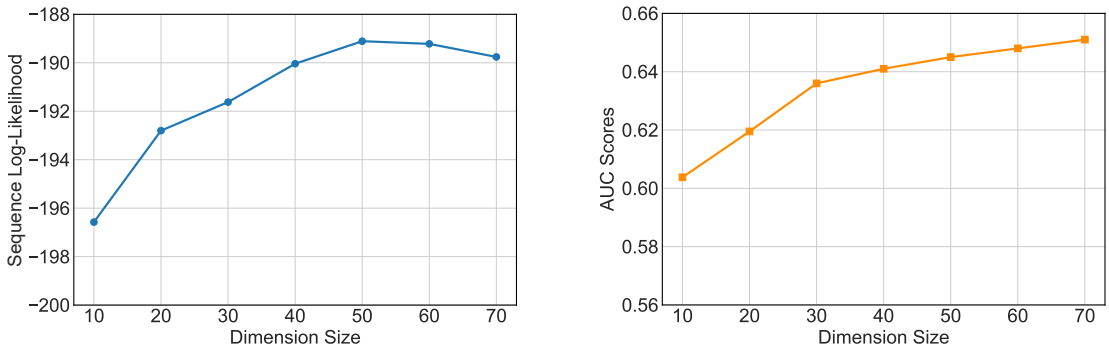
Datasets	Methods	50%	60%	70%	80%	90%
Weibo	Single	0.534	0.536	0.540	0.545	0.546
	Role-Based	**0.645**	**0.648**	**0.652**	**0.654**	**0.655**
Twitter	Single	0.604	0.607	0.610	0.611	0.613
	Role-Based	**0.680**	**0.682**	**0.689**	**0.694**	**0.698**

training cascades increases. This proves the ineffectiveness of embedding concatenation. On the contrary, the AUC scores of the proposed DNRL keeps higher than its variant NRL and rises obviously with the increase of training cascades ratio. This indicates that the joint representation learning framework makes a good use of diffusion information to facilitate the link prediction. Additionally, it is found that the diffusion-only variant DRL performs gradually better and even outperforms NE-EIC when more than 40% cascades of Weibo dataset are given. This demonstrates that the diffusion cascades do carry important clues for missing link inference.

5.3.4 Other Discussions

Effect of Role-Based Representation

In order to testify the effect of role-based representation, we conduct a further comparative experiment. We compare the performance of role-based representations with that of single representation for each user. The experimental results on diffusion prediction and link prediction are shown in Table 5.4. The role-based representations



(a) Diffusion Prediction Results (b) Link Prediction Results

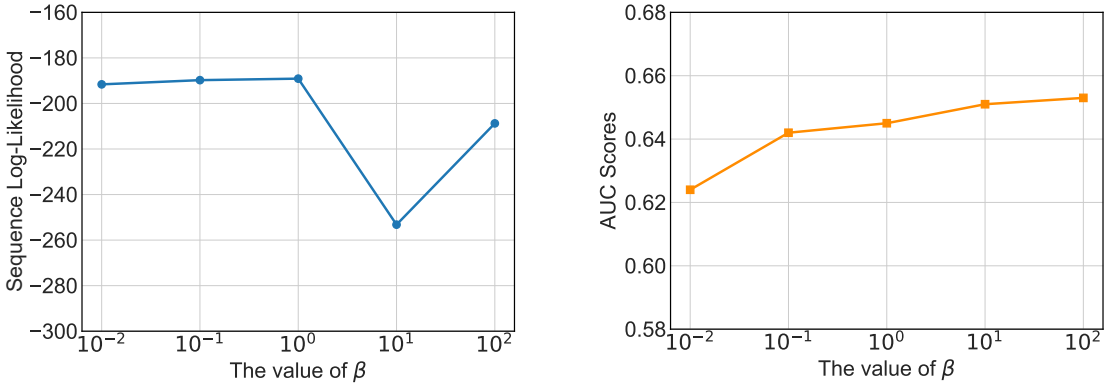
Figure 5.6: Performance w.r.t Dimension Size

consistently achieve better performance than single representation on both tasks. This is because that the role-based representations can clearly differentiate the effect of user roles. Modeling user characteristics from role-based view can help to provide more accurate features on different aspects of users.

Parameter Sensitivity

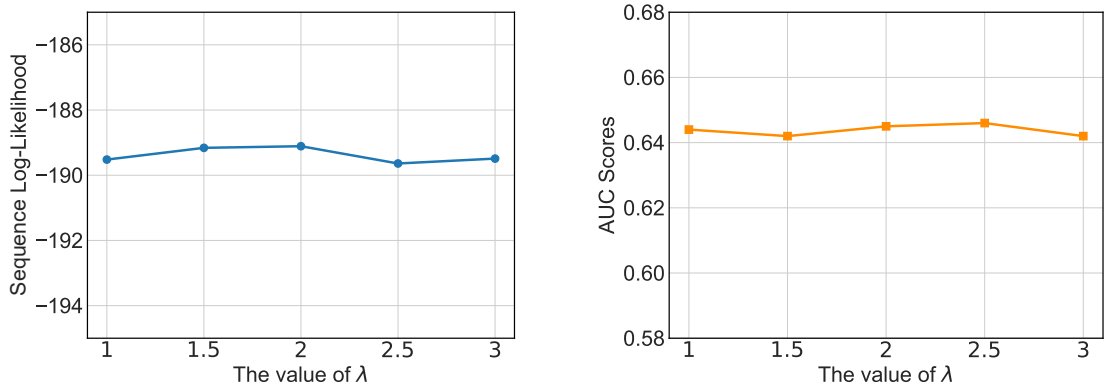
We take the Weibo dataset as an example to test the sensitivity of parameters in the proposed model.

Dimensionality is the most important parameter in the representation learning method, thus we firstly investigate the effect of dimension size in the proposed model. We conduct experiments with different size of user representations, while keeping other parameters fixed. As illustrated in Figure 5.6, the dimension size of user representations does affect the performance on both tasks, but the impacts have different trends. As for diffusion prediction, the performance boosts until the dimension size reaches 50 and tends to be steady after that point; while for link prediction, the dimensionality continuously brings positive influence on the performance and the improvement gradually slows down when the size exceeds 30. In order to balance the performance on both two tasks, we experimentally set the dimension size as 50



(a) Diffusion Prediction Results (b) Link Prediction Results

Figure 5.7: Performance w.r.t β



(a) Diffusion Prediction Results (b) Link Prediction Results

Figure 5.8: Performance w.r.t λ

in all above experiments.

Another hyper parameter is the β, which is used to control the scale of latent distance. To testify the impact of β, we select different magnitudes for β and other parameters are fixed (dimension size is 50) in this experiment. As shown in Figure 5.7, the link prediction benefits from the increase of β's magnitude, but the diffusion prediction will receive a negative impact if β is larger than 1. To avoid this large decrease of diffusion prediction performance, it is suggested to set β as 1.

The last parameter is the λ, which is used to control the scale of diffusion influence. From the results shown in Figure 5.8, we can conclude that the value of λ has small influence on the two tasks. In the diffusion prediction evaluation, a normalization is conducted on the predicted probability, which largely reduces the impact of diffusion influence. In practice, we only set this parameter as 2 to rescale the learned diffusion influence in [0,1].

5.4 Chapter Summary

In this chapter, we have explored the correlation between information diffusion and relationship building behaviors on social media. To capture the correlation, we proposed to embed user characteristics of both diffusion cascades and social network into a shared representation space. A unified user representation learning model is developed, which defines two learning objectives on two behaviors respectively. An efficient algorithm is proposed to jointly optimize the two objectives. Thanks to the generative modeling properties of both learning objectives, the proposed model can be directly applied to diffusion prediction and link prediction tasks. We conduct experiments on two real social media datasets, i.e., Weibo and Twitter. The proposed model outperforms state-of-the-art methods on both tasks. In further analysis, the proposed model shows better robustness when reducing training data of one behavior. This proves that the proposed model effectively leverages the correlated effect.

Part II

Neural Network Based Models

Chapter 6

A Sequential Neural Information Diffusion Model with Structure Attention

6.1 Chapter Overview

Most conventional diffusion prediction methods assumed that diffusion cascades followed some prior diffusion mechanisms, such as independent cascade model or linear threshold model. Although recently proposed representation learning methods unlocked some limitations of graph-based models, most of them still required for an underlying diffusion mechanism. The effectiveness of these methods heavily relied on the hypothesis of the underlying diffusion mechanisms, which is hard to specify or verify in practice [139].

Since most retrievable diffusion cascades are recorded as sequences, recent researches began to formulate the problem as the sequence prediction task and developed sequential models for the problem, which aimed to circumvent the above problem without an underlying diffusion mechanism. Sequential models focus on modeling how historical diffusion information affects the future diffusion behavior. For instance, Manavoglue et al. [97] proposed a user behavior generation method based on maximum entropy and Markov mixture model. Most recently, with the

great success of recurrent neural network (RNN) in sequence modeling, a series of RNN-based sequential models were proposed for diffusion prediction and achieved better performance than other non-neural-network based sequential approaches, as claimed by Wang et al. [139]. In particular, Du et al. [38] were the first to propose a RNN framework to model and predict cascade, where timing and mark information are embedded to parameterize the generation process of cascades. Based on this framework, Wang et al. [139] proposed a sequence-to-sequence model, which additionally employed a machine translation alignment mechanism. There are mainly two benefits of RNN-based sequence modeling. It is able to avoid strong prior assumption of underlying diffusion mechanism, and it is flexible to capture sequential dependencies or memory effect in diffusion cascades.

However, most existing RNN-based models failed to take available structure information of user graph into account. In the literature, user connection graphs have proved crucial for understanding the actual diffusion dynamics [114, 112, 29]. Apparently, connections among users are the fundamental communication channel. Without the structural information restriction and regulation, the sequential models alone are not able to identify and predict the direction of information flow accurately. This may result in poor generalization ability if training cascades are insufficient or biased. Therefore, it is non-trivial to explore how to effectively integrate both sequential and structural information for diffusion prediction.

In this work, we develop a novel **S**equential **N**eural **I**nformation **D**iffusion model with **S**tructure **A**ttention (**SNIDSA**). It explores both diffusion sequences and user connection graph. Specifically, the proposed model employs a RNN-based framework to model the historical sequential diffusion. Meanwhile, in order to incorporate the structural diffusion context, we propose a Structure Attention Module (SAM), which builds upon the structure attention mechanism to model the potential future diffusion directions through user communication channels. To effectively integrate

the sequential and structural information, a gating mechanism is designed for neural hidden state updating. The prediction ability and the robustness of SNIDSA are verified on four synthetic datasets with different network structures and diffusion patterns and a real diffusion dataset. The effectiveness of the proposed structure attention and gating mechanism are further demonstrated by ablation studies.

The main contributions of this work are summarized as follows.

- We propose a novel RNN-based sequential model SNIDSA for diffusion prediction. This model is able to comprehensively capture both sequential properties in diffusion cascades and structural information in user graph.
- We design a structure attention and a gating mechanism to effectively explore possible structural context and integrate sequential and structural information for diffusion prediction.
- We evaluate the proposed model on both synthetic datasets and real-life dataset. It outperforms state-of-the-art RNN-based sequential models. The further ablation studies also demonstrate the effectiveness of proposed structure attention and gating mechanism.

6.2 Method

6.2.1 Basic RNN for Diffusion Prediction

We firstly introduce the basic RNN model for diffusion sequence modeling. RNN is a feed-forward neural network, which can be used to predict a cascade sequentially. As shown in Figure 6.1, at step i, the i-th infected user u_i is embedded into a vector $\mathbf{x}_i$ as input. The input is fed into hidden units of RNN by nonlinear transformation $f(\cdot)$, jointly with the outputs from the last hidden units (hidden states $\mathbf{h}_{i-1}$), updating the hidden state $\mathbf{h}_i = \text{RNN}(\mathbf{x}_i; \mathbf{h}_{i-1})$. The hidden state $\mathbf{h}_i$ is used to encode the

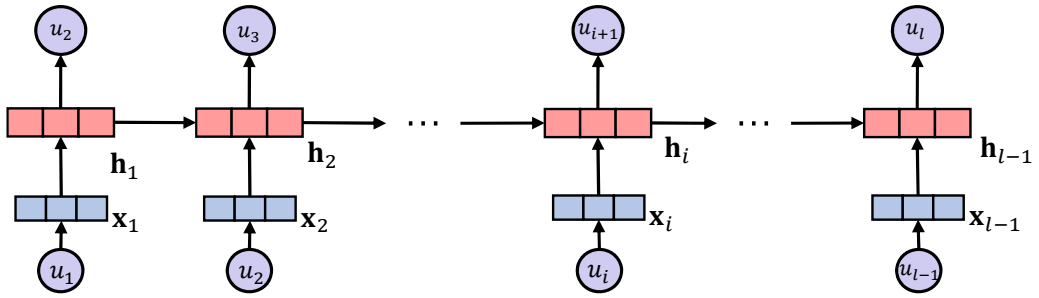


Figure 6.1: Basic RNN Framework for Diffusion Prediction

information of historical infected users and the model is trained to predict next infected user u_{i+1} given $\mathbf{h}_i$. Therefore, RNN aims to maximize the likelihood of cascade c as follows:

$$p(c) = \prod_{i=1}^{l-1} p(u_{i+1}|(u_1, ..., u_i)) = \prod_{i=1}^{l-1} p(u_{i+1}|\mathbf{h}_i) \tag{6.1}$$

The basic RNN updates the hidden state with following formula:

$$\mathbf{h}_i = \sigma(\mathbf{W}_h \mathbf{x}_i + \mathbf{U}_h \mathbf{h}_{i-1} + \mathbf{b}_i) \tag{6.2}$$

where $\sigma()$ represents the non-linear function. It has proven that the basic RNN is not able to capture long-term dependencies in sequence. To make up this disadvantage, some RNN variants with gating mechanisms were proposed. The gates are used to control the information flow when updating hidden states, which makes it possible to keep long-term historical information in hidden state. Here we introduce two widely used RNN variants, i.e., LSTM (Long Short-Term Memory) [61] and GRU (Gated

Recurrent Unit) [31]. The LSTM updates the hidden state as follows:

$$\begin{aligned}
\mathbf{f}_i &= \sigma_g(\mathbf{W}_f \mathbf{x}_i + \mathbf{U}_f \mathbf{h}_{i-1} + \mathbf{b}_f) \\
\mathbf{r}_i &= \sigma_g(\mathbf{W}_r \mathbf{x}_i + \mathbf{U}_r \mathbf{h}_{i-1} + \mathbf{b}_r) \\
\mathbf{o}_i &= \sigma_g(\mathbf{W}_o \mathbf{x}_i + \mathbf{U}_o \mathbf{h}_{i-1} + \mathbf{b}_o) \\
\mathbf{c}_i &= \mathbf{f}_i \odot \mathbf{c}_{i-1} + \mathbf{r}_i \odot \sigma_c(\mathbf{W}_c \mathbf{x}_i + \mathbf{U}_c \mathbf{h}_{i-1} + \mathbf{b}_c) \\
\mathbf{h}_i &= \mathbf{o}_i \odot \sigma_h(\mathbf{c}_i)
\end{aligned} \tag{6.3}$$

A LSTM unit is composed of a cell $\mathbf{c}_i$, a forget gate $\mathbf{f}_i$, an input gate $\mathbf{r}_i$ and an output gate $\mathbf{o}_i$. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell. The computations of LSTM may not be very efficient in practice. The GRU was proposed to simplify the computations of gates and has fewer parameters than LSTM. The GRU updates the hidden state as follows:

$$\begin{aligned}
\mathbf{z}_i &= \sigma_g(\mathbf{W}_z \mathbf{x}_i + \mathbf{U}_z \mathbf{h}_{i-1} + \mathbf{b}_z) \\
\mathbf{r}_i &= \sigma_g(\mathbf{W}_r \mathbf{x}_i + \mathbf{U}_r \mathbf{h}_{i-1} + \mathbf{b}_r) \\
\mathbf{h}_i &= (1 - \mathbf{z}_i) \odot \mathbf{h}_{i-1} + \mathbf{z}_i \odot \sigma_h(\mathbf{W}_h \mathbf{x}_i + \mathbf{U}_h(\mathbf{r}_i \odot \mathbf{h}_{i-1}) + \mathbf{b}_h)
\end{aligned} \tag{6.4}$$

A GRU does not maintain a cell and only contains two gates, i.e., forget gate $\mathbf{z}_i$ and reset gate $\mathbf{r}_i$, to control the information flow.

Based on sufficient training cascades, RNN models are able to find an optimal solution for the conditional probability of next infected user $p(u_{i+1}|(u_1, ..., u_i))$, avoiding the bias on underlying diffusion mechanisms. Thus, RNN offers us a promising and flexible method to capture the complex propagation patterns in diffusion cascade modeling.

6.2.2 The Proposed SNIDSA Model

Model Overview

The proposed **SNIDSA** model is illustrated in figure 6.2. The model employs an RNN framework to model the information diffusion process sequentially, which maintains a hidden state to memorize the summarized diffusion history. At each time step t_i, the infected user u_i is represented as a low dimensional vector through a mapping matrix $\mathbf{\Phi} \in \mathbb{R}^{N \times d_x}$. The user embedding vector is represented as $\mathbf{x}_i = \mathbf{\Phi}[u_i]$ with the dimension d_x. In addition to the infected user, we also consider u_i's structural diffusion context, $\mathbf{s}_i$, which is captured through a Structure Attention Mechanism (SAM) over u_i's neighbors (either direct or indirect) in the user graph $\mathbf{A}$. Given the infected user $\mathbf{x}_i$ and its structural diffusion context $\mathbf{s}_i$, the model updates the hidden state sequentially. A gating mechanism is utilized to effectively integrate $\mathbf{x}_i$ and $\mathbf{s}_i$. Like LSTM, we also introduce two types of hidden states, i.e., internal state represented as $\mathbf{c}$ and output state denoted by $\mathbf{h}$. At step i, the gate computes output state jointly based on previous internal state, user embedding and its structure context, i.e., $\mathbf{h}_i = Gate(\mathbf{x}_i, \mathbf{s}_i, \mathbf{c}_{i-1})$. Given the output state $\mathbf{h}_i$, the SNIDSA model then predicts the probability of next infected user as follow:

$$\hat{p}(u_{i+1}|\mathbf{h}_i) = \text{softmax}(\mathbf{W}_h \mathbf{h}_i + \mathbf{b}_h) \tag{6.5}$$

where $\mathbf{W}_h \in \mathbb{R}^{N \times d_h}, \mathbf{b}_h \in \mathbb{R}^N$. The learning objective is to maximize the likelihood of all infected users in a diffusion sequence:

$$\max_{\Theta} \prod_{i=1}^{l-1} p(u_{i+1}|\mathbf{h}_i) \tag{6.6}$$

where Θ represents the set of parameters in SNIDSA.

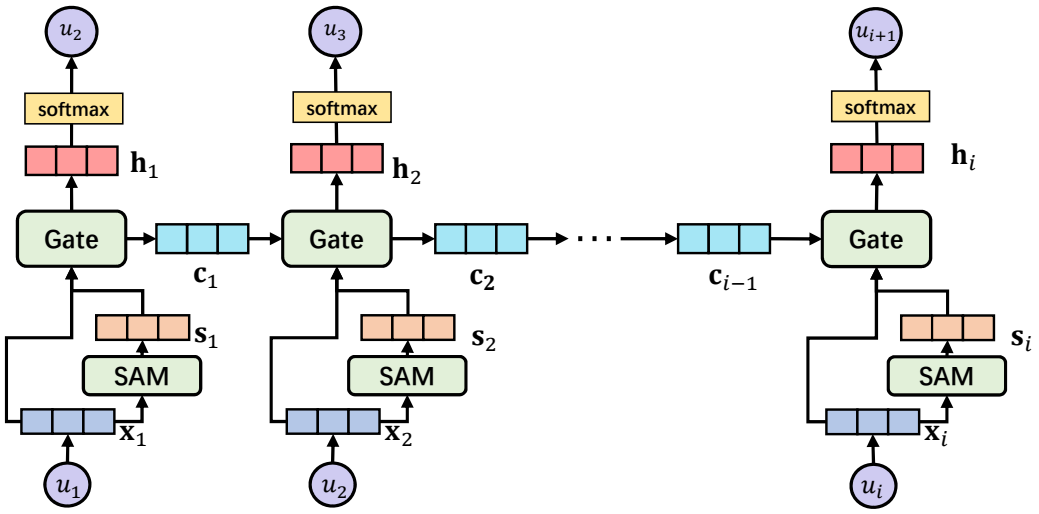


Figure 6.2: Overview of SNIDSA Model

Structure Attention

The Structure Attention Module (SAM) of SNIDSA is developed to model the structural diffusion context for each infected user. It formulates the regulation of communication channels on the potential diffusion directions from the current infected user by exploiting the user graph $\mathbf{A}$. Based on the learned attention, the structural context vector $\mathbf{s}_i$ is constructed. The illustration of SAM is provided in Figure 6.3. The user graph is taken as the external knowledge for attention computation.

Given user embeddings, we define a scoring function, which measures the diffusion potential from user u_i to user u_j as follow:

$$a_{ij} = g(u_i, u_j) \cdot \mathbf{w}^T[\mathbf{x}_i; \mathbf{x}_j] \tag{6.7}$$

where u_j, who is connected to u_i directly or indirectly, is a structural context user of u_i. $g(\cdot)$ is a pairwise proximity function defined over the user graph, which measures how closely two users are connected in the graph. $\mathbf{w}^T[\mathbf{x}_i; \mathbf{x}_j]$ is a concatenation-based attention mechanism [133], which measures the attention score of u_i to each u_j, where $w \in \mathbb{R}^{2d_x}$ and $[;]$ means the vector concatenation.

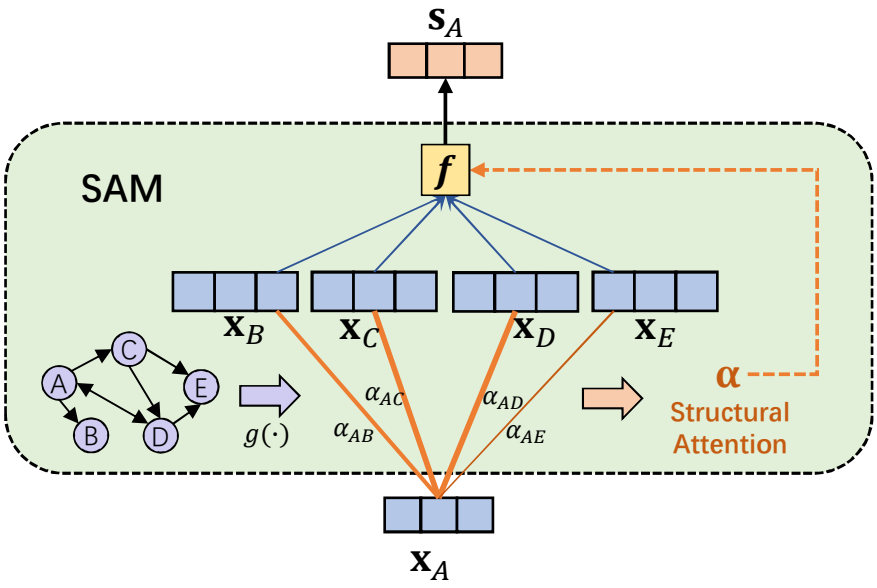


Figure 6.3: Structure Attention Module

To make the attention score comparable across different users, we apply the softmax normalization function. For simplicity, let α denote the normalized attention score, then it is computed as:

$$\alpha_{ij} = \frac{\exp(a_{ij})}{\sum_{k=1,k\neq i}^{N} \exp(a_{ik})} \tag{6.8}$$

Given the structure attention scores of all context users, and their user embeddings, the structural context vector $\mathbf{s}_i$ for each infected user u_i is constructed by weighted sum pooling.

$$\mathbf{s}_i = f(\sum_{j=1,j\neq i}^{N} \alpha_{ij}\mathbf{x}_j) \tag{6.9}$$

where $f(\cdot)$ is a non-linear activation function.

Gating Mechanism

Taking both the current infected user $\mathbf{x}_i$, and his/her structural context $\mathbf{s}_i$ as the input, SNIDSA iteratively updates the hidden state at each time step. We design a

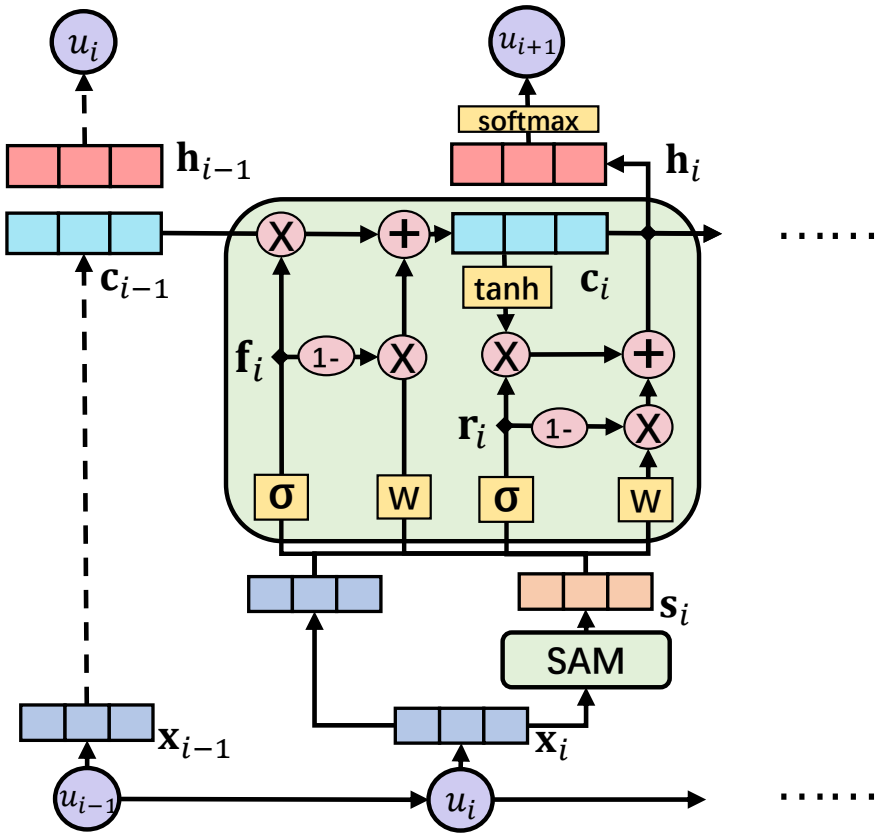


Figure 6.4: Gating Mechanism in SNIDSA

gating mechanism to integrate the historical diffusion information compressed in the hidden state, the current infected user and his/her diffusion context. The proposed gate is illustrated as Figure 6.4. Specifically, the proposed gate decomposes the hidden state into two different states. The internal state $\mathbf{c}_i$ is used to memorize the diffusion history while the output state $\mathbf{h}_i$ is used to predict the next potential infected user. Correspondingly, two gates, i.e., a forget gate and a reset gate, are designed to modulate the two states respectively. The detailed updating formulas are as follows:

$$\mathbf{f}_i = \sigma(\mathbf{W}_f \mathbf{x}_i + \mathbf{U}_f \mathbf{s}_i + \mathbf{b}_f) \tag{6.10}$$

$$\mathbf{r}_i = \sigma(\mathbf{W}_r \mathbf{x}_i + \mathbf{U}_r \mathbf{s}_i + \mathbf{b}_r) \tag{6.11}$$

$$\mathbf{c}_i = \mathbf{f}_i \odot \mathbf{c}_{i-1} + (1 - \mathbf{f}_i) \odot (\mathbf{W}_c \mathbf{x} + \mathbf{U}_c \mathbf{s}_i) \tag{6.12}$$

$$\mathbf{h}_i = \mathbf{r}_i \odot \tanh(\mathbf{c}_i) + (1 - \mathbf{r}_i) \odot (\mathbf{W}_h \mathbf{x}_i + \mathbf{U}_h \mathbf{s}_i) \tag{6.13}$$

where σ denotes the sigmoid function; $\mathbf{W}_f, \mathbf{W}_r, \mathbf{W}_c, \mathbf{W}_h, \mathbf{U}_f, \mathbf{U}_r, \mathbf{U}_c, \mathbf{U}_h \in \mathbb{R}^{d_h \times d_x}$ and $\mathbf{b}_f, \mathbf{b}_r \in \mathbb{R}^{d_h}$. The forget gate $\mathbf{f}_i$ is used to update the internal state $\mathbf{c}_i$, which aims to forget the irrelevant part of historical information and keep the important part of the latest information; The reset gate $\mathbf{r}_i$ is used to compute the new output state $\mathbf{h}_i$ based on internal state $\mathbf{c}_i$ and the linear combination of $\mathbf{x}_i$ and $\mathbf{s}_i$.

Model Learning

We apply the negative log-likelihood function to define the loss function on the training dataset $C = \{c_1, ..., c_M\}$ as:

$$\mathcal{L}(C) = -\sum_{m=1}^{M} \sum_{i=1}^{l_m - 1} \log p(u_{i+1} | \mathbf{h}_i) \tag{6.14}$$

where u_{i+1} is the infected user in diffusion sequence c_m at time step t_{i+1}, l_m is the length of sequence c_m. The learning objective is to minimize the above loss function. The back-propagation through time (BPTT) algorithm is applied for training. The parameters are updated by Adam optimizer with mini-batch.

6.3 Experiments

6.3.1 Compared Models and Experiment Setups

We compare the proposed SNIDSA model with a set of state-of-the-art diffusion models, including a graph-based model, a representation learning model and RNN-based models.

- **CTIC** [114]: Independent Cascade (IC) model is the mainstream graph-based information diffusion model. It assumes that the information diffuses along

the edges existing in the user graph and learns a diffusion probability for each edge. CTIC is a continuous-time version of IC model, which considers both interpersonal influence and diffusion delay pattern, and achieves state-of-the-art performances as reported in previous work [18].

- **EIC** [18]: Embedded Independent Cascade Model also grounds on the IC schema. Instead of learning probabilities on a discrete graph, EIC models the diffusion space as a continuous latent space where relative positions of users are used to define the diffusion probabilities. EIC is able to infer diffusion influence for all pairs of users without knowing a given discrete graph.

- **RNN**: We consider the vanilla RNN as a baseline sequential model.

- **LSTM** [61]: Long Short Term Memory (**LSTM**) network employ a complex gating mechanism based on basic RNN, which is able to capture long-term dependencies. Compared with other gated recurrent models, LSTM is found to perform consistently better, thus only LSTM is compared in experiments.

- **RMTPP** [38]: Recurrent marked temporal point process (RMTPP) is a state-of-the-art sequential models for sequence prediction. Besides modeling marker (diffusion user) sequence, it additionally models timing information with a temporal point process.

- **CYAN-RNN** [139]: It is the latest sequence-based neural diffusion model and achieves better performance than other popular sequential models as claimed in [139]. Similar to our SNIDSA, it models information diffusion as a sequence. Differently, it employs an encoder-decoder framework to map original diffusion sequence to a sequence of next users at each timestep, and uses an alignment mechanism to capture dependencies among users.

All these models are evaluated on the next infected user prediction task. Due to the large number of potential targets, the prediction task is often cast as the retrieval (ranking) problem [18, 139]. Specifically, each model outputs the infection probability distribution over all users and the ground-truth infected user is expected to get a highest probability. Hence, two widely adopted ranking metrics are used for evaluation. They are Mean Reciprocal Rank (MRR) and Accuracy on top k ($A@k$) [139].

The parameter settings are as follows. The dimension sizes of user embedding and hidden state for all embedding-based and neural models are 32 and 64, respectively. Other parameters follow their recommended settings in published papers. As for SNIDSA, the learning rate is 0.001 and the batch size is 32. We assume the first-order proximity of graph in Equation 2, i.e., $g(\cdot) = 1$ if the element of adjacency matrix $A_{ij} = 1$, and $g(\cdot) = 0$ otherwise. The activation function $f(\cdot)$ used for structural context composition in Equation 4 is Exponential Linear Unit (ELU) [32].

6.3.2 Experiments on Synthetic Dataset

The experiments on synthetic data aim to validate the effectiveness of the proposed SNIDSA model in the diffusion prediction task concerning different types of network structures and diffusion patterns.

Data Generation

Following previous work [139, 112], we generate synthetic data in two steps, i.e., network generation and diffusion simulation. We apply Kronecker Graph Model [83] to generate two types of networks with different structures. One is Random Network (RD) with the parameter matrix [0.5 0.5; 0.5 0.5]. The other is Hierarchical Community Network (HC) with parameters [0.9 0.1; 0.1 0.9]. Both are widely used in previous diffusion studies. As the result, we construct two graphs with 2^n users,

Table 6.1: Statistics of Synthetic Data Sets

	RD-Exp	RD-Ray	HC-Exp	HC-Ray
# Train Cascades	16,000	16,000	16,000	16,000
# Dev. Cascades	2,000	2,000	2,000	2,000
# Test Cascades	2,000	2,000	2,000	2,000
# Edges	1,021	1,023	714	716
Cascade Length	70.50	57.33	16.96	16.17

where we use default setting $n = 9$. We then use the generative approach [112], which follows the IC schema, to simulate information diffusion on each generated graph. We select two representative transmission patterns described in the previous work [112], i.e., Exponential and Rayleigh, to generate diffusion sequences. We set the maximum diffusion time as 100. Finally, we end up with 4 synthetic datasets. The detailed statistics are presented in Table 6.1. We randomly sample 80% diffusion sequences as training data, and take the rest as validation and testing data with an even split.

Evaluation Results

The experimental results are shown in Table 6.2. Overall, the proposed SNIDSA model outperforms all the other methods concerning different network structures and diffusion patterns. It suggests the following conclusions.

- Structure information is important for diffusion prediction. The proposed SNIDSA and the graph-based method CTIC, both of which utilize the given user connection information, achieve relatively better performance than other compared methods. Although EIC grounds similar schema with CTIC, it is too difficult to accurately infer diffusion influence of all user pairs in latent space without knowing the underlying user graph.

Table 6.2: Diffusion Prediction Performance on Synthetic Data (%)

Model	HC-Exp			HC-Ray			RD-Exp			RD-Ray		
	MRR	A@5	A@10	MRR	A@5	A@10	MRR	A@5	A@10	MRR	A@5	A@10
CTIC	63.36	82.92	92.92	60.73	81.72	92.68	37.71	**51.06**	61.17	28.62	38.09	53.83
EIC	40.78	50.25	59.17	38.92	43.94	51.73	10.64	18.32	21.56	9.21	13.58	15.49
RNN	64.58	80.84	87.65	62.15	77.22	86.20	29.54	39.10	49.22	19.29	25.63	35.84
LSTM	65.71	82.99	91.95	65.08	83.74	89.78	36.78	49.45	60.13	34.33	47.55	59.87
RMTPP	65.12	81.23	89.90	63.97	81.55	88.10	33.89	45.26	57.74	31.24	45.32	56.11
CYANRNN	49.71	60.88	72.17	43.24	49.38	61.84	20.13	29.26	38.12	12.57	20.19	28.92
SNIDSA	**67.19**	**86.40**	**93.20**	**70.58**	**88.93**	**94.64**	**37.72**	50.48	**62.13**	**37.77**	**51.51**	**64.59**

- RNN framework is capable of modeling the sequential diffusion process. Overall, RNN-based sequential models achieve good performance in the experiments. Surprisingly, the state-of-the-art neural sequential diffusion model CYAN-RNN performs even worse than baseline sequential models. CYAN-RNN formulates diffusion prediction as a sequence-to-sequence problem. We doubt that transferring single diffusion sequence modeling to a sequence-to-sequence learning violates the nature of diffusion process.

- The proposed SNIDSA effectively captures and integrates useful structure information in the sequential diffusion prediction framework. This can be demonstrated by the improvement of SNIDSA over CTIC. Different from CTIC that models diffusion process completely on the graph structure, the SNIDSA treats the graph as the constraint in diffusion sequence modeling and the proposed structure attention mechanism provides a more effective and flexible way to utilize structure information.

6.3.3 Experiments on Real Data

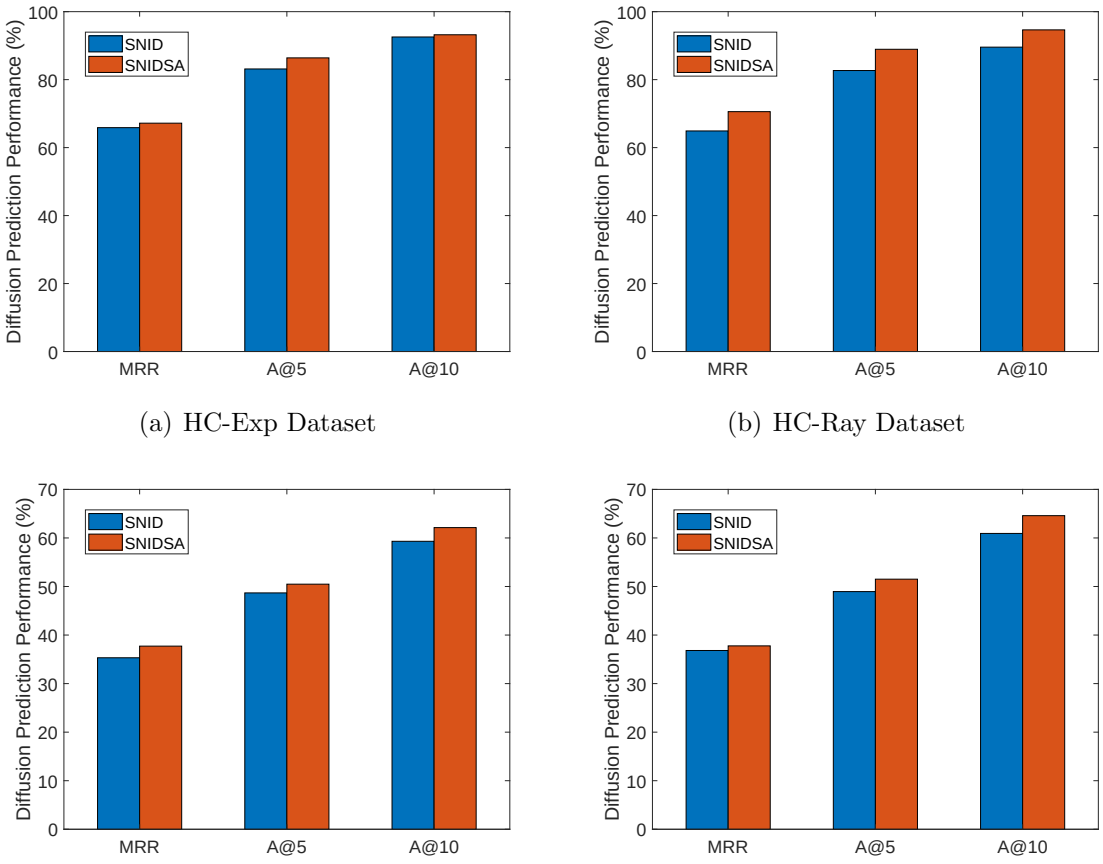
To further demonstrate the effectiveness of the proposed model, we conduct comparative experiments on a real diffusion dataset, i.e., MemeTracker [82]. This dataset contains articles from mainstream news websites or blogs. Each piece of diffusion data records a diffusion process of a specific key phrase and is represented by a sequence of websites with timestamps. An element in the sequence represents which website writes an article mentioning the key phrase at what time. Following the previous work [18], we construct the connection graph according to the citation relation between websites. That is, if website A has at least one article citing an article published by B, an edge is linked from B to A. Frequent websites (users) and their corresponding citations (diffusion) are selected. The final dataset contains 1109

Table 6.3: Diffusion Prediction Performance on Real Data(%)

Model	MRR	A@5	A@10
CTIC	7.80	9.96	12.54
EIC	7.60	8.23	10.77
RNN	23.26	31.17	40.23
LSTM	24.08	32.76	41.49
RMTPP	23.35	31.79	41.37
CYANRNN	10.63	15.24	21.97
SNIDSA	**26.17**	**35.53**	**45.79**

nodes, 31823 edges, 33992 training diffusion sequences, 4250 validation sequences and 4250 testing sequences.

The experimental results on real data are shown in Table 6.3. Similar to the results on synthetic data, SNIDSA achieves the best performance consistently. It is worth noting that the performance of the graph-based model CTIC drops drastically compared with the results on synthetic data, while SNIDSA remains robust across both datasets. Since CTIC holds strong assumption that message can only propagate along the given graph structure, the performance heavily depends on the accurateness and completeness of graph information. In this real dataset, even though there are no mis-linked edges, it is still difficult to guarantee the information of constructed graph is complete, e.g., the communication connections between websites are not limited in citation relationships. In the contrast, the SNIDSA model employs the attention mechanism, which is able to selectively inject useful structure information. This improves the robustness of the proposed model.



(a) HC-Exp Dataset

(b) HC-Ray Dataset

(c) RD-Exp Dataset

(d) RD-Ray Dataset

Figure 6.5: The Effect of Structure Attention (SNID v.s SNIDSA)

6.3.4 Analysis

The Effect of Structure Attention

In order to testify whether the proposed attention mechanism can effectively capture useful structure information and bring improvement for the sequential modeling, we compare SNIDSA with its structure-free version, denoted as **SNID**. In essence, SNID is a recurrent sequential model where the user graph is ignored.

We conduct comparative experiments on the above generated synthetic datasets. The results are reported in Figure 6.5. SNIDSA consistently outperforms its structure-free variant SNID on the four synthetic datasets in terms of all evaluation metrics.

This remarkably indicates the importance of structure attention in the proposed model. The structure attention module effectively extracts important structural context of historical users. It provides useful information of potentially infected users based on graph structure in addition to historical information in the diffusion sequence. This could reduce the prediction space of next infected user, therefore, the prediction accuracy is improved.

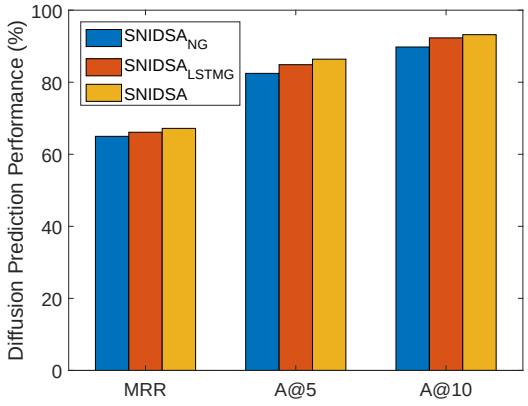
The Effect of Proposed Gate

The gate in SNIDSA is proposed to integrate the extracted structural information into the sequential hidden state. To demonstrate the effectiveness of the proposed gating mechanism, we further conduct a comparative experiment on the synthetic datasets. In this experiment, we compare the performance of SNIDSA with the following two variants:

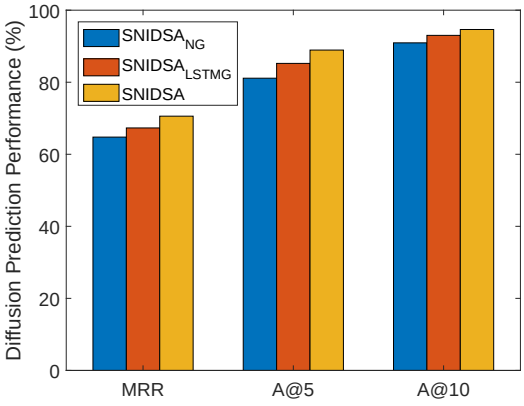
- $\text{SNIDSA}_{\text{NG}}$: This variant removes the proposed gate, and the hidden state is updated like the basic RNN. The user embedding $\mathbf{x}$ and structural context $\mathbf{s}$ are concatenated and fed to hidden state.

- $\text{SNIDSA}_{\text{LSTMG}}$: This variant replaces the proposed gate with the complex gate of LSTM, in which user embedding $\mathbf{x}$ and structural context $\mathbf{s}$ are also concatenated as the input of recurrent cell of LSTM.

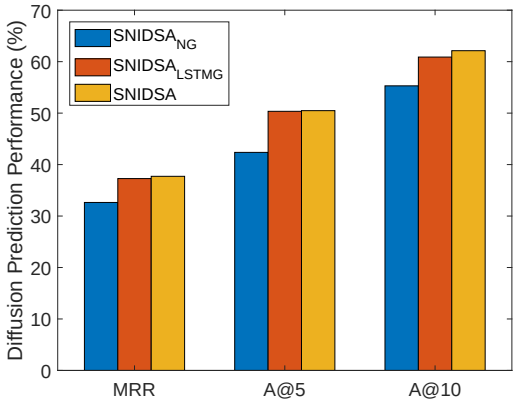
As shown in Figure 6.6, SNIDSA consistently achieves better performance than the two variants on the four datasets. Without the fusion gate, the variant $\text{SNIDSA}_{\text{NG}}$ performs worse than other two gate-based methods. This indicates that it is more reasonable to use gating mechanism to fuse the sequential and structural information rather than integrating them with simple concatenation and linear combination. Additionally, SNIDSA performs better than the variant $\text{SNIDSA}_{\text{LSTMG}}$. The gating mechanism in LSTM achieves better than other baseline sequential models. This



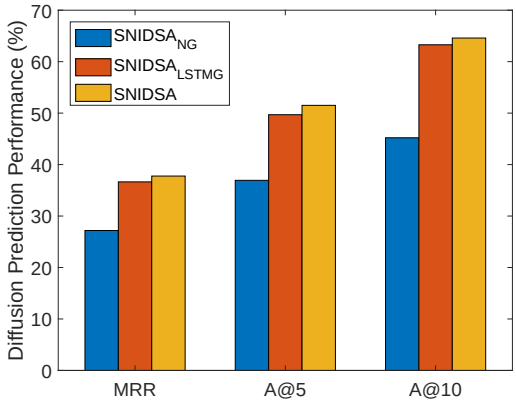
(a) HC-Exp Dataset



(b) HC-Ray Dataset



(c) RD-Exp Dataset



(d) RD-Ray Dataset

Figure 6.6: The Effect of Proposed Gating Mechanism

shows that the proposed gating mechanism is more effective for information fusion than popular gates. The special design of the proposed gate is able to selectively capture important parts and drop unimportant parts in sequential and structural information.

6.4 Chapter Summary

In this chapter, we propose a sequential neural network that well incorporates structure attentions for information diffusion modeling. In particular, the RNN-based

framework is employed to capture the sequential patterns of historical information diffusion while the attention mechanism formulates a user's structural potential diffusion context with a social graph. An effective gating mechanism is then designed to integrate both sequential and structural properties behind the diffusion process for recurrent state updating. When evaluated on the diffusion predication task, the effectiveness of the proposed model is demonstrated on both synthetic and real datasets. The further ablation studies also show the proposed structure attention and gating mechanism are effective to extract useful structural context and fuse structural and sequential information for diffusion prediction, respectively.

Chapter 7

Hierarchical Diffusion Attention Network

7.1 Chapter Overview

As mentioned in previous chapter, due to the sequential form of retrievable information diffusion cascades, researchers recently formulated the problem as a sequence prediction task: given the historically infected users in an information cascade, the next infected user is predicted. Inspired by the great success of recurrent neural network (RNN) in sequence modeling, a series of RNN-based sequential models were proposed and their effectiveness was demonstrated on the real diffusion data [38, 139, 136]. These models sequentially encode the historical information as hidden states and predict next infected user based on the compressed states. However, the real diffusion process behind sequential cascades does not strictly follow the sequential assumption. This is because there exists an underlying user connection graph, which may not be explicitly unobserved but can directly determine the diffusion dependencies among users. For example, given a cascade $\{(A, t_A), (B, t_B), (C, t_C), (D, t_D)\}$ and an underlying graph as shown in Figure 7.1, the sequential models assume that the infections of C and D are influenced by the hidden states h_2 (compressed information of A and B) and h_3 (compressed infor-

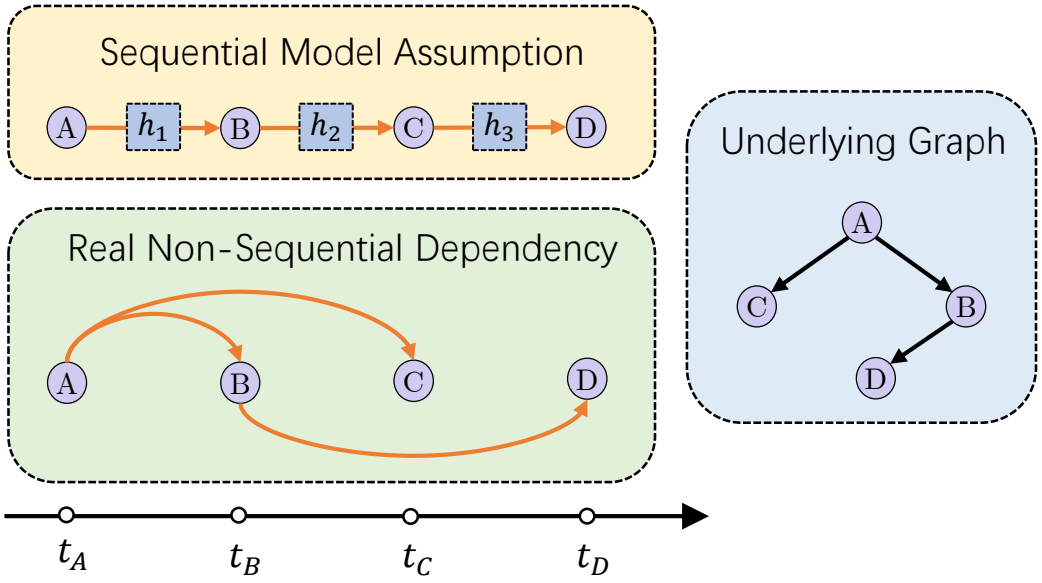


Figure 7.1: An Example of Non-Sequential Diffusion Dependency

mation of A, B and C) respectively. But in fact, C and D are directly dependent on A and B according to the graph structure. This kind of non-sequential dependency has also been identified as an important characteristic of diffusion sequences in the previous work [139]. Though the gating mechanisms in existing sequential models [61] can selectively drop the information of B from hidden state h_2 when generating C, they also lead to the loss of dependency of D on B in hidden state h_3. The hidden states of the compressed information are not expressive enough for such non-sequential diffusion dependency, thus the prediction power is limited.

To this end, we propose a **Hi**erarchical **D**iffusion **A**ttention neural **N**etwork (**HiDAN**) for the problem of predicting diffusion when the underlying graph is unknown. To capture unique properties of diffusion sequences, we devise a non-sequential architecture with two-level attention mechanisms. Specifically, a user-level dependency attention is suggested to dynamically capture diffusion dependencies among historical users. A fusion gate is then designed to selectively integrate user's self-information and its dependency context. Based on the dependency-aware

historical user information, a cascade-level influence attention, which considers both inherent importance and time-decay effects, is developed to infer the influence of historical users on potentially infected future users. The inferred influence can be interpreted as the possible dependencies of the future user on all historical users. We evaluate the proposed model against state-of-the-art sequential diffusion prediction models on three real diffusion datasets. The significantly better performance demonstrates the effectiveness of our model. The case studies on synthetic datasets further indicate that the learned dependency attentions are mostly consistent with the underlying graph. In addition, HiDAN also shows higher efficiency than sequential models in experiments. The main contributions of this work are summarized as follows:

- To the best of our knowledge, we are the first to develop a **non-sequential neural network framework** for diffusion prediction problem, which is well-adapted to properties of real diffusion cascades.
- We propose two-level attention mechanisms for cascade modeling, i.e., a **user-level dynamic dependency attention**, which effectively captures historical diffusion dependencies, and a **cascade-level time-aware influence attention**, which infers future dependencies by modeling user inherent importance and time-decay effects.
- The experiments on three real datasets demonstrate the significantly improved **effectiveness** and **efficiency** of the proposed model compared with state-of-the-art approaches.

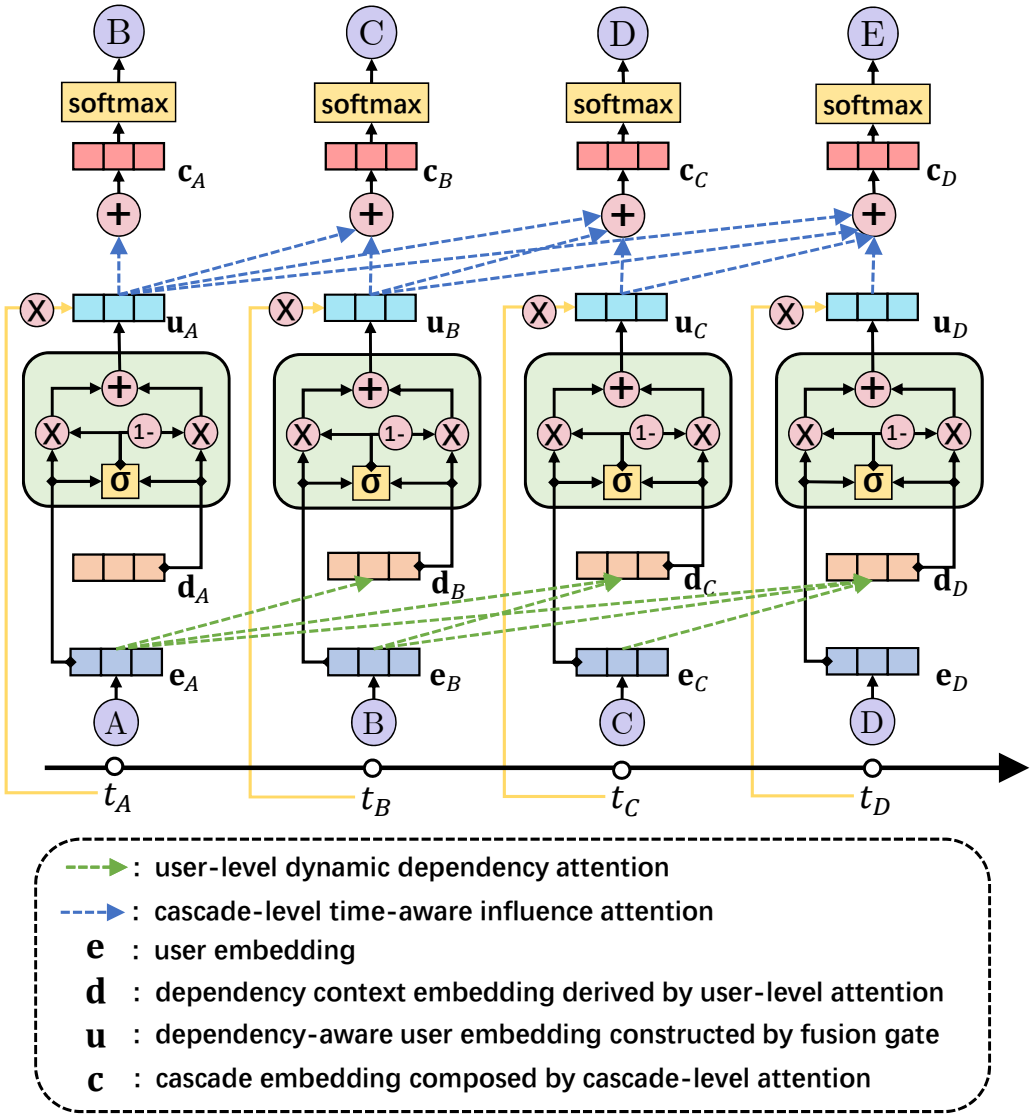


Figure 7.2: Overview of HiDAN Model

7.2 Method

7.2.1 The HiDAN Model

The framework of the proposed HiDAN is illustrated in Figure 7.2. Initially, each user of an input cascade is embedded as a user embedding. Given the embedded user

information, the user-level attention mechanism dynamically captures the diffusion dependencies between each user and its context users. A gating mechanism is developed to integrate a user's own information and his/her dependency context. Based on the dependency-aware user information, the cascade-level attention computes the influence of historical users on possible future users by capturing users' inherent importance and the time-decay effects. Given the cascade embedding constructed with the influence attention, the model then predicts the next infected user.

User Embedding

At time t_i, the sequence of already infected users $\{u_1, ..., u_i\}$, ordered by infection time, is regarded as the input to the model. The raw representation of each input user $u_j \in \{u_1, ..., u_i\}$ is the one hot vector of user ID, i.e., $\mathbf{x}_j \in \mathbb{R}^N$, where N is the total number of distinct users. To extract expressive high-level features of users, we transform the raw input $\mathbf{x}$ to the user embedding $\mathbf{e}$ via a fully-connected layer:

$$\mathbf{e}_j = f_x(\mathbf{W}_x \mathbf{x}_j + \mathbf{b}_x) \tag{7.1}$$

where $\mathbf{W}_x \in \mathbb{R}^{d \times N}$, $\mathbf{b}_x \in \mathbb{R}^d$ are learnable parameters, d is the size of the embedding and f_x is the non-linear activation function.

User-Level Dynamic Dependency Attention

This attention mechanism aims at capturing diffusion dependencies among input cascade users and extracting dependency-aware user features. The diffusion dependency describes who possibly infect(s) whom in the diffusion process, which possesses the following two characteristics. (1) Each cascade user can only be infected by its previous users, thus the dependency of u_j only exists on $\{u_1, ..., u_{j-1}\}$. The previously infected users $\{u_1, ..., u_{j-1}\}$ are referred to as the diffusion context users of u_j. (2) Diffusion dependency is directional, i.e., the high dependency of u_j on u_k does not

indicate same level of dependency of u_k on u_j. This is mainly because the dependency relationship is often directed in the real user graph. For example, in Twitter, a star user, who is followed by millions of users, frequently infects his/her followers instead of being infected by followers. Therefore, it requires differentiating different roles that users play in the directed dependencies.

Based on the above understanding, we propose a dynamic attention mechanism to capture the diffusion dependency for each input user. The dependency attention score between user $u_j \in \{u_1, ..., u_i\}$ and its context user $u_k \in \{u_1, ..., u_{j-1}\}$ is measured as follows:

$$\alpha_{kj} = \frac{\exp(\langle \mathbf{W}_e^c \mathbf{e}_k, \mathbf{W}_e^t \mathbf{e}_j \rangle)}{\sum_{l=1}^{j-1} \exp(\langle \mathbf{W}_e^c \mathbf{e}_l, \mathbf{W}_e^t \mathbf{e}_j \rangle)} \tag{7.2}$$

where $\mathbf{W}_e^c, \mathbf{W}_e^t \in \mathbb{R}^{d \times d}$ are transformation matrices for the context user and the target user respectively; $\langle \cdot, \cdot \rangle$ represents the inner product. $\mathbf{W}_e^c, \mathbf{W}_e^t$ are employed to differentiate user roles in directed dependencies. When checking whether u_j is dependent on u_k, u_k is regarded as the context user and transformed by $\mathbf{W}_e^c$, while u_j is treated as the target user and transformed by $\mathbf{W}_e^t$. When checking dependency with the opposite direction, the roles of u_k and u_j are reversed, thus dependency scores are different. Similar to most attention mechanism, we apply a softmax function to derive the probability distribution. Therefore, α_{kj} denotes the probability that u_j is dependent on u_k over all his/her context users.

The useful context information of u_j, denoted as $\mathbf{d}_j$, is computed via the following attention weighted sum:

$$\mathbf{d}_j = \sum_{k=1}^{j-1} \alpha_{kj} \mathbf{e}_k \tag{7.3}$$

The above dependency attention process for one specific user is illustrated in Figure 7.3. Since dependency attentions and context embeddings are computed independently for each input user in the proposed non-sequential framework, we are

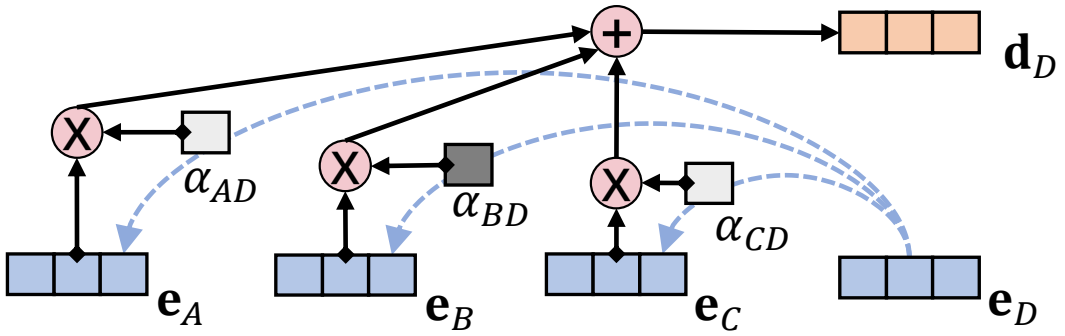


Figure 7.3: Dependency Attention Mechanism: An Example for u_D

able to parallelize the dynamic processes as matrix computation. Given a input cascade c, where the cascade length is l and the embedding matrix of l users is $\mathbf{E} \in \mathbb{R}^{d \times l}$, we replace time-consuming enumeration by using a mask matrix $\mathbf{M} \in \mathbb{R}^{l \times l}$, where each entry $M_{i,j} = 0$ if $i < j$; otherwise $M_{i,j} = -\infty$. Then we derive the matrix of attentions as:

$$\mathbf{A} = \text{softmax}((\mathbf{W}_e^c \mathbf{E})^T (\mathbf{W}_e^t \mathbf{E}) + \mathbf{M}) \tag{7.4}$$

where $\mathbf{A} \in \mathbb{R}^{l \times l}$. Each row vector $\boldsymbol{\alpha}_j$ in $\mathbf{A}$ represents u_j's attentions on its context users. The mask forces the softmax function to compute valid attentions only over u_j's context users and assign 0 to other users (infected later than u_j). The matrix of context embeddings can be derived as: $\mathbf{D} = \mathbf{A}\mathbf{E}^T$.

Dependency-Aware Fusion Gating

For each input cascade user u_j, we now have user embedding $\mathbf{e}_j$ and his/her diffusion context embedding $\mathbf{d}_j$. To selectively integrate the important information of two embeddings, a concise and effective fusion gating mechanism is employed. It produces a dependency-aware user representation $\mathbf{u}_j$ as follows:

$$\mathbf{g}_j = \text{sigmoid}(\mathbf{W}_g^1 \mathbf{e}_j + \mathbf{W}_g^2 \mathbf{d}_j + \mathbf{b}_g) \tag{7.5}$$

$$\mathbf{u}_j = \mathbf{g}_j \odot \mathbf{e}_j + (1 - \mathbf{g}_j) \odot \mathbf{d}_j \tag{7.6}$$

where $\mathbf{W}_g^1, \mathbf{W}_g^2 \in \mathbb{R}^{d\times d}$ and $\mathbf{b}_g \in \mathbb{R}^d$. $\mathbf{g}$ is used to drop unimportant parts of user embedding $\mathbf{e}_j$ and add new information of its diffusion context embedding $\mathbf{d}_j$ such that the fused embedding $\mathbf{u}_j$ is aware of the diffusion dependency.

Cascade-Level Time-Aware Influence Attention

At the cascade level, we also consider the non-sequential dependency, in which all historical users could trigger the future infection with different probabilities. We interpret such future dependencies as dynamic influence of historical users on the whole cascade, and propose a time-aware influence attention mechanism. Based on the inferred influence, we compose dependency-aware embeddings $\mathbf{u}$ to cascade-level features for final prediction. This attention mechanism captures two factors: the inherent importance of users to cascade and the dynamic time-decay effects.

The inherent importance of u_j describes how important the information in the dependency-aware embedding $\mathbf{u}_j$ is to the cascade. If only considering the inherent importance, we can define the influence with the self-attention mechanism [92] as: $\langle \mathbf{w}, f_u(\mathbf{W}_u\mathbf{u}_j + \mathbf{b}_u)\rangle$.

However, user influence is generally assumed to decrease as time passes. This is known as the time-decay effect. Empirical studies [112] have shown that time-decay patterns in different data are not identical, thus predefining the form of time-decay function is often impractical [22]. We estimate the time-decay factor in HiDAN via a neural function directly. For $u_j \in \{(u_1, t_1), ..., (u_i, t_i)\}$, the past time at current step t_i can be represented as $\Delta t^j = t_i - t_j$. We discretize the information of past time as a one-hot vector $\text{vec}(\Delta t^j) = \mathbf{t}^j \in \mathbb{R}^T$, where $\mathbf{t}_n^j = 1$ if $t_{n-1} < t_i - t_j < t_n$. The critical time points of the discretization, such as t_{n-1} and t_n, are defined by splitting the time range $(0, T_{\max}]$ into T intervals $\{(0, t_1], ...(t_{n-1}, t_n], ..., (t_{T-1}, T_{\max}]\}$, where $T_{\max}$ is the max observation time. We aim at mapping the past time $\mathbf{t}^j$ to a vector $\boldsymbol{\lambda}_j$, describing latent features of time-decay. To capture the non-linearity of the decay

effect, we compute $\boldsymbol{\lambda}_j$ via the following fully connected layer:

$$\boldsymbol{\lambda}_j = \text{sigmoid}(\mathbf{W}_t \mathbf{t}^j + \mathbf{b}_t) \tag{7.7}$$

where $\mathbf{W}_t \in \mathbb{R}^{d \times T}$ and $\mathbf{b}_t \in \mathbb{R}^d$.

Taking both inherent importance and time decay factors into consideration, we define the following time-aware influence attention:

$$\beta_j = \frac{\exp(\langle \mathbf{w}, \boldsymbol{\lambda}_j \odot f_u(\mathbf{W}_u \mathbf{u}_j + \mathbf{b}_u) \rangle)}{\sum_{k=1}^{i} \exp(\langle \mathbf{w}, \boldsymbol{\lambda}_k \odot f_u(\mathbf{W}_u \mathbf{u}_k + \mathbf{b}_u) \rangle)} \tag{7.8}$$

where $\mathbf{W}_u \in \mathbb{R}^{d \times d}$, $\mathbf{b}_u \in \mathbb{R}^d$ and $\mathbf{w} \in \mathbb{R}^d$. The decay factor vector $\boldsymbol{\lambda}_j$ serves as a soft gate, which selectively drops the information of already infected users according to their infection times.

Given the user influence β_j, this layer finally composes all dependency-aware user embeddings and constructs the cascade embedding at time t_i as follows:

$$\mathbf{c}_i = \sum_{j=1}^{i} \beta_j \mathbf{u}_j \tag{7.9}$$

Prediction Layer

Given cascade embedding $\mathbf{c}_i$ at t_i, HiDAN predicts the probability of next infected user over all possible users as:

$$\hat{p}(u_{i+1} | \mathbf{c}_i) = \text{softmax}(\mathbf{W}_c \mathbf{c}_i + \mathbf{b}_c) \tag{7.10}$$

where $\mathbf{W}_c \in \mathbb{R}^{N \times d}$, $\mathbf{b}_c \in \mathbb{R}^N$.

7.2.2 Model Learning

Given the training set $C = \{c^1, ..., c^M\}$, the learning objective is to minimize the following negative log-likelihood loss:

$$\mathcal{L}(C) = -\sum_{m=1}^{M} \sum_{i=1}^{n_m - 1} \log \hat{p}(u_{i+1} | \mathbf{c}_i^m) \tag{7.11}$$

where u_{i+1} is truly infected user in cascade c^m at time t_{i+1}. The backpropagation algorithm is utilized in the training process. As for parameters updating, we employ stochastic gradient descent (SGD) method with mini-batch and adopt the Adam optimizer [70].

7.3 Experiments

7.3.1 Data

To verify the effectiveness of the proposed model, we conduct comparative experiments on the following three real datasets, which are representative in information diffusion studies.

- **MemeTracker** [82]: This dataset contains articles from mainstream news websites or blogs. Each cascade records the diffusion process of a specific key phrase and is represented by a sequence of webpage links associated with corresponding timestamps.

- **Weibo** [158]: This dataset consists of content reposting logs crawled from Sina Weibo, a Chinese microblogging site. Each reposting log represents a diffusion process, in which users are ordered as a sequence according to the time they repost.

- **Twitter** [148]: This dataset records the diffusion processes of hash-tags in Twitter. The sequences of users and timestamps of using the same hash-tags are traced as diffusion cascades.

Following the previous work [139], we select frequent users and corresponding cascades as experimental data. The detailed statistics are presented in Table 7.1. We randomly sample 80% of cascades for training and the rest for validating and testing with an even split.

Table 7.1: Statistics of Experimental Data

	MemeTracker	Weibo	Twitter
# Users	1,109	8,190	13,755
# Cascades	42,492	43,365	72,103
Avg. Cascade Length	8.8	22.5	9.4

7.3.2 Baselines

We compare the proposed model, HiDAN, with the following popular and strong sequential baselines.

- **RNN**: RNN represents the basic recurrent neural network sequential model.
- **LSTM** [61]: Long short-term memory (LSTM) network is a stronger RNN-based sequential model, which employs an effective gating mechanism to control the information flow in sequence.
- **RMTPP** [38]: Recurrent marked temporal point process (RMTPP) is the state-of-the-art sequential models for sequence prediction. Besides modeling marker (diffusion user) sequence, it additionally models timing information with a temporal point process.

The following state-of-the-art attention based sequential models are compared. All of them compute attentions on hidden states.

- **Att-RNN**: Att-RNN employs a representative attention mechanism [96] in RNN. Attentions are computed between current hidden state and previous states.
- **Att-LSTM**: Att-LSTM employs the same attention mechanism as Att-RNN in the LSTM framework.

Table 7.2: Diffusion Prediction Performance (%)

	MemeTracker				Weibo				Twitter			
Model	*MRR*	*A@10*	*A@50*	*A@100*	*MRR*	*A@10*	*A@50*	*A@100*	*MRR*	*A@10*	*A@50*	*A@100*
RNN	23.26	40.23	66.33	77.24	1.33	2.25	6.04	9.49	2.04	3.68	9.83	14.92
LSTM	24.08	41.49	67.23	77.92	1.40	2.63	7.23	11.49	2.47	4.69	11.78	16.63
RMTPP	23.35	41.37	66.34	76.99	1.35	2.28	6.69	10.73	1.73	3.17	8.96	13.64
Att-RNN	23.51	42.05	67.14	78.10	1.57	2.52	7.51	12.18	2.53	4.56	13.68	20.14
Att-LSTM	24.39	43.11	68.69	79.55	1.64	2.93	8.20	12.60	2.73	5.08	14.78	21.71
CYANRNN	17.62	35.84	57.29	69.81	1.06	1.53	5.18	7.83	1.19	1.82	5.69	8.97
HiDAN	**27.91**	**48.89**	**74.63**	**84.44**	**2.48**	**4.30**	**11.31**	**17.30**	**5.74**	**11.18**	**23.61**	**30.41**

- **CYAN-RNN** [139]: This is the latest attention-based sequential method for cascade prediction. Instead of using a single-chain RNN, it employs a encoder-decoder architecture where a coverage-based alignment mechanism is applied. Attentions are computed between current decoder states and previous encoder states.

7.3.3 Evaluation Metrics and Settings

The performance is evaluated by predicting the next infected user based on previous infections. Due to the large number of potential targets, this prediction task is often regarded as a ranking problem [139]. Given the output probabilities of all users, the ground-truth user, who is truly infected at next step, is expected to get higher probability. We adopt two widely used ranking metrics for evaluation: Mean Reciprocal Rank (MRR) and Accuracy on top k ($A@k$) [139].

The size of hidden unit is set to 64 for all baselines. Other parameters of baselines follow the recommended settings in original papers. For the proposed models, the dimension size of d is also 64, the learning rate is 0.001, the max observation time $T_{\max}$ is 120 hours, the number of splitting time interval T is 40, and the non-linear activation functions f_x, f_u are selected as Exponential Linear Unit (ELU) [32]. We also apply the Dropout [125] with the keep probability 0.8 and the L2 regularization on parameters to avoid over-fitting.

7.3.4 Evaluation Results

As shown in Table 7.2, the proposed HiDAN consistently and remarkably outperforms all compared methods in terms of MRR, A@10, A@50 and A@100 on three datasets. The superiority of HiDAN on diffusion prediction is clearly demonstrated. In addition, we observe the following important findings.

- **The non-sequential framework is capable for cascade modeling**. Hi-

DAN significantly outperforms all baselines. Instead of maintaining hidden states sequentially, HiDAN utilizes attention-based non-sequential layers to memorize historical infection information. This indicates that the proposed non-sequential architecture is capable of modeling cascade without sequential assumptions.

- **Attention mechanism is beneficial for diffusion prediction**. Almost all attention-based sequential models perform better than their non-attention versions. The proposed hierarchical attentions are specific for capturing historical non-sequential dependencies and inferring future dependencies. Attentions in sequential models also aim at computing long-term dependencies to alleviate sequential assumptions. This finding is consistent with our argument that modeling non-sequential dependencies is important for diffusion prediction.

- **HiDAN is more effective in capturing diffusion dependencies**. Compared with state-of-the-art attention-based sequential models, HiDAN gains a very significant improvement. The attention-based sequential models define attentions on hidden states, which represent the accumulated information but not independent information of each historical user. They cannot clearly capture user-to-user dependencies. Differently, HiDAN does not sequentially compress user information but directly computes user-level attentions with unique user embeddings. HiDAN captures dependencies more explicitly and effectively.

7.3.5 Analysis and Discussion

Ablation Studies

To evaluate the contributions of different components, we further conduct ablation studies, which consider following variants of HiDAN:

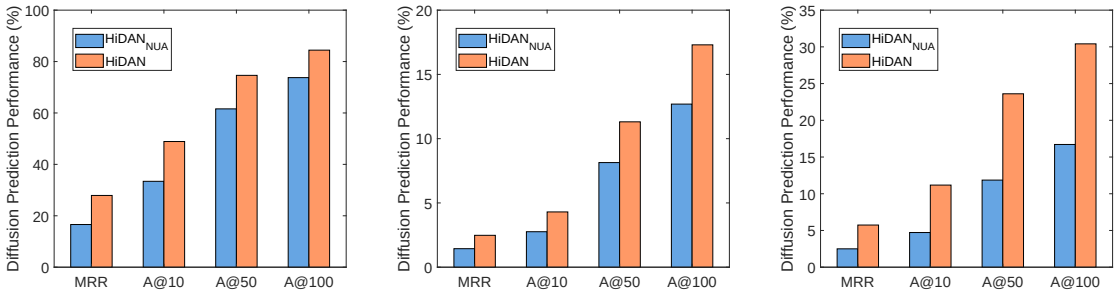
- **HiDAN$_{\mathbf{NUA}}$**: This is a user-level attention-free variant, which replaces dependency attention and fusion gate with an average operation over embeddings of context users.

- **HiDAN$_{\mathbf{NCA}}$**: This is a cascade-level attention-free variant. Influence attention is substituted by the average operation.

- **HiDAN$_{\mathbf{NT}}$**: This is a variant without considering time-decay in cascade-level attention.

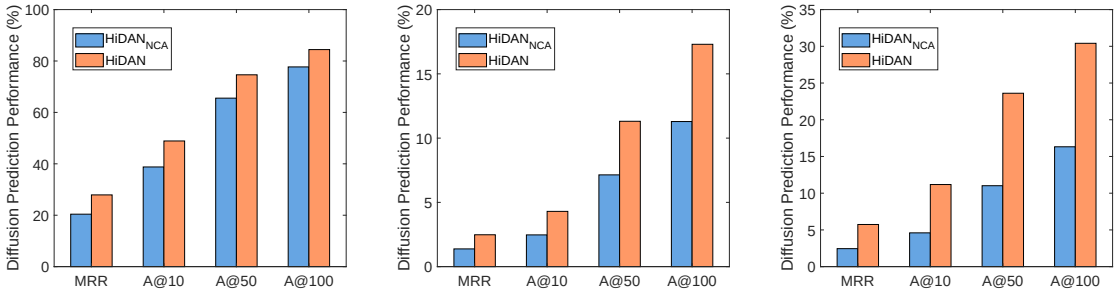
As shown in Figure 7.4, we first compare the performance of HiDAN against the variant HiDAN$_{\text{NUA}}$. It is found that when the user-level attention is removed, the diffusion prediction performance drops down drastically and consistently on all datasets. This remarkably demonstrates the contribution of the proposed user-level attention mechanism to HiDAN model. Previous experiments have shown that learning diffusion dependencies are significantly important for diffusion prediction. With the meticulous design based on special characteristics of cascades, this user-level attention mechanism is able to explicitly capture user-to-user diffusion dependencies among historically infected users. Based on accurate dependencies, the model can encode more useful historical context information while dropping out unimportant parts. Therefore, the prediction ability of the model is improved. In next subsection, we will present some case studies to further illustrate the performance of the proposed dependency attention.

Furthermore, we compare HiDAN with the variant HiDAN$_{\text{NCA}}$. The results are illustrated in Figure 7.5. Similar to above experiment, HiDAN outperforms HiDAN$_{\text{NCA}}$ significantly and consistently on all datasets. This obviously indicates that the cascade-level attention also plays a crucial role in our HiDAN model. The aim of this cascade-level attention mechanism is inferring the influence of historically infected users on possible future infections. The higher influence of one historical user



(a) MemeTracker Dataset (b) Weibo Dataset (c) Twitter Dataset

Figure 7.4: The Effect of User-Level Dependency Attention



(a) MemeTracker Dataset (b) Weibo Dataset (c) Twitter Dataset

Figure 7.5: The Effect of Cascade-Level Influence Attention

represents s/he has higher possibility to be the trigger of next infected user. Thus, the influence is concerned with possible dependencies in the future. With comprehensively modeling user inherent importance and time-decay effect, this mechanism is effective to infer user influence on future infections.

Another ablation study is to testify the effect of neural time-decay function in the proposed cascade-level attention. In this ablation study, HiDAN is compared with the variant HiDAN_{NT}. The results are reported in Table 7.3. We can find that ignoring time-decay effect in influence attention brings consistently negative impact on the proposed model. This suggests that time decay matters to influence modeling. Taking time information into account helps infer user influence (dependencies) on

Table 7.3: The Effect of Neural Time-Decay Function

Data	Model	MRR	A@10	A@50	A@100
MemeTracker	HiDAN_{NT}	27.12	47.92	73.47	83.26
	HiDAN	**27.91**	**48.89**	**74.63**	**84.44**
Weibo	HiDAN_{NT}	2.39	4.06	11.02	16.83
	HiDAN	**2.48**	**4.30**	**11.31**	**17.30**
Twitter	HiDAN_{NT}	5.46	11.05	23.08	29.12
	HiDAN	**5.74**	**11.18**	**23.61**	**30.41**

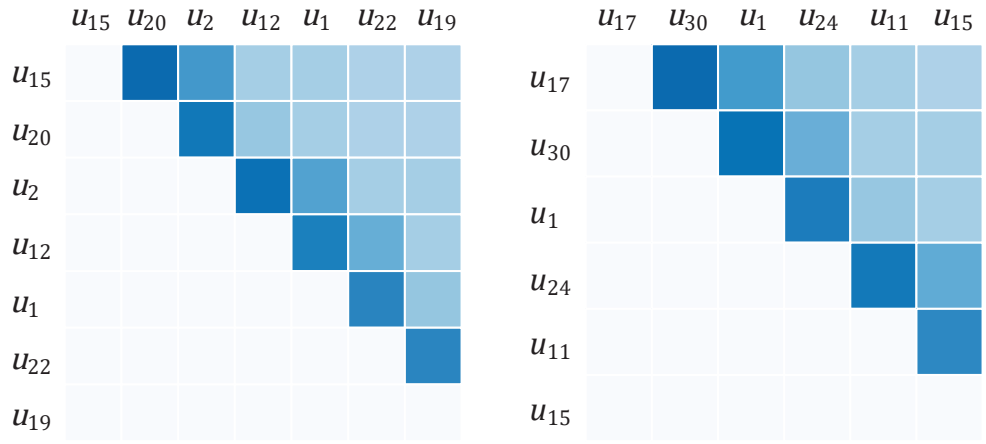
future infections more accurately.

Case Studies on Diffusion Dependency

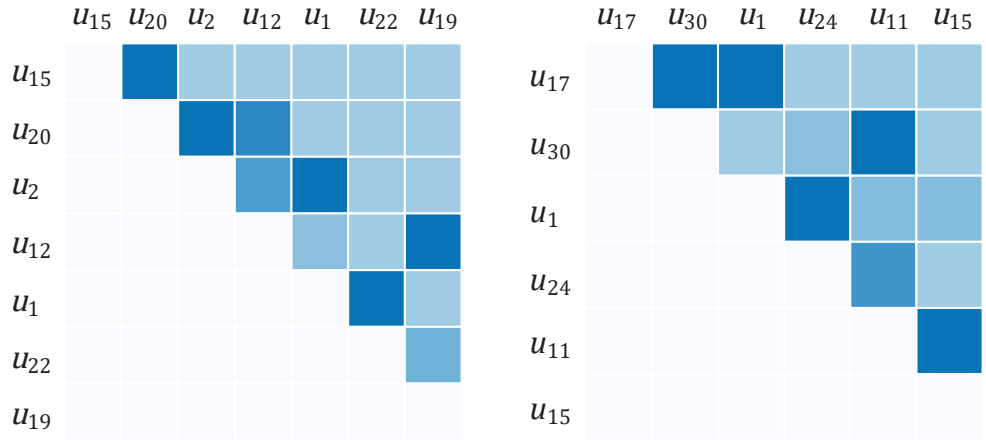
The experimental results have shown that user-level dependency attention plays an important role in HiDAN. Here, we further investigate whether the proposed mechanism is better at capturing user-to-user diffusion dependencies. Since it is very difficult to access the complete graph of real data, we utilize two synthetic data (CP-Exp and RD-Exp) provided in the previous work [139] for case studies. The graphs are created by Kronecker Generator [83]. Given the created graph, cascades are generated by simulation processes [112].

We visualize attention matrices of the sampled cases learned by HiDAN and its best competitor Att-LSTM. Meanwhile, the adjacency matrices of the users who are involved in the cases are visualized as ground-truth. As illustrated in Figure 7.6, each element (u_i, u_j) in the learned attention matrices indicates how much u_j is infected by u_i. The deeper the color is, the greater the attention is. Each element (u_k, u_l) in the ground-truth matrices denotes whether or not there is a directed edge from u_k to u_l (i.e., black indicates 'yes' and gray indicates 'no').

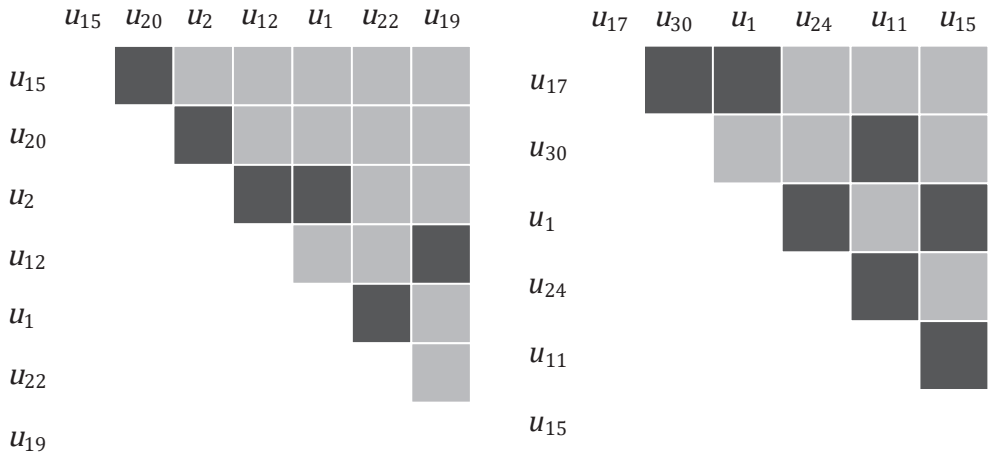
Compared with Att-LSTM, attentions learned by HiDAN are more consistent with ground-truth dependencies. Despite of employing attention mechanism, Att-



(a) Dependencies Learned by Att-LSTM



(b) Dependencies Learned by HiDAN



(c) Ground-Truth Dependencies

Figure 7.6: Case Studies on Learned Dependencies

Table 7.4: Average Training Time (seconds) per Epoch

	# Param.	MemeTracker	Weibo	Twitter
RNN	6.2k	22 s	145 s	261 s
LSTM	24.8k	34 s	190 s	346 s
RMTPP	8.8k	24 s	148 s	265 s
Att-RNN	14.5k	29 s	152 s	273 s
Att-LSTM	33.1k	38 s	203 s	422 s
CYAN-RNN	27.1k	$\geqslant$22 s	$\geqslant$145 s	$\geqslant$261 s
HiDAN	25.6k	**12 s**	**36 s**	**85 s**

LSTM mainly focuses on the most recent hidden state. On the contrary, HiDAN is more aware of cross dependencies, especially long-term and multi- dependencies. In the sampled case of the CP-Exp data (left 3 images), the dependencies of all users except u_{12} are correctly allocated by HiDAN. In the sampled case of the RD-Exp (right 3 images), HiDAN accurately captures dependencies for u_{30}, u_1 and u_{24}. It is interesting that both u_{11} and u_{15} have multiple paths connected to previously infected users, as shown in the ground-truth matrix. HiDAN is able to recognize such kind of multiple diffusion paths.

Efficiency Analysis

Apart from effectiveness, higher efficiency is another crucial advantage of HiDAN. To demonstrate this, we conduct a training time comparison. All models except CYAN-RNN[1] are implemented with Tensorflow and trained on the same GTX1080Ti graphic card with the same batch size.

As shown in Table 7.4, HiDAN has relatively fewer parameters than Att-LSTM and CYAN-RNN. More importantly, HiDAN is super faster than all compared se-

[1]Released code is in JAVA. GPU training time is unknown. But as an RNN-based model, it is at least not faster than RNN.

quential models. This can be attributed to its non-sequential architecture. The recurrent layer in sequential models has an approximate $O(ld^2)$ complexity. However, HiDAN replaces the recurrent layer with the dependency attention, which can be parallelized with matrix computation as shown in Equation 7.4, and time complexity is only about $O(l^2d)$. Compared with hidden size d (64 in experiments), average cascade length l (9, 23 and 10 in experiments) is often smaller in real data. Therefore, HiDAN has a lower complexity at the historical user encoding layer. Additionally, without sequential assumptions, HiDAN can compute outputs of all steps in parallel with a $O(1)$ complexity at the prediction layer, whereas sequential models need to output step by step with a $O(l)$ complexity. These dramatically speed up the training of HiDAN especially when the cascade length is getting larger.

7.4 Chapter Summary

In this chapter, we propose a novel hierarchical attention neural network for diffusion prediction, which is well adapted to the non-sequential characteristics of diffusion cascades. The proposed two-level attentions are able to capture historical user-to-user dependencies and infer future dependencies. The experiments on three real diffusion datasets demonstrate the effectiveness and efficiency of our model when compared with state-of-the-art sequential models. The further analysis and case studies illustrate the important contributions of the hierarchical attention mechanism.

Chapter 8

Conclusions and Suggestions for Future Research

In the past decade, social media has brought revolutionary changes on the way that information propagates among individuals. Massive information diffusion processes are triggered and recorded in social media. These observed diffusion processes provide rich sources for a variety of potentially valuable applications. For example, companies are able to optimize their marketing strategies through forecasting the diffusion of advertisements and governments can maintain stability by tracking the diffusion of public opinions. All these related applications require for a good understanding of diffusion mechanisms and accurate predictions of diffusion dynamics. This presents unprecedented challenges and opportunities for researches on information diffusion, driving many researchers, in recent years, to study the diffusion phenomena in social media and particularly focus on the **diffusion prediction** problem, which studies how users participate and affect other users in information diffusion and predicts the future diffusion process with the learned knowledge.

In this thesis, we comprehensively study the problem of information diffusion prediction by proposing and solving three important sub-problems, i.e., how to build diffusion prediction model only based on historical diffusion processes, how to utilize observed network information to improve the generalization ability of the diffusion

prediction model, and how to capture the interplay between diffusion process and network structure to predict dynamics of diffusion and network simultaneously. Based on the two powerful machine learning frameworks, i.e., representation learning and neural network, we proposed a series of models for these three problems. Compared with state-of-the-art studies, our proposed models consistently obtain significant improvements on single diffusion prediction task or joint diffusion and link prediction tasks.

8.1 Summary of Contributions

The following sections summarize main contributions of this thesis.

8.1.1 Network Regularized User Representation Learning

- Different from most previous graph-based studies, we propose to project diffusion users as role-based representations to capture latent user-specific characteristics.
- We are the first to integrate network structure information in diffusion user representation learning and develop a novel network regularized representation learning model to learn role-based user representations.
- The experiments on three real-life datasets significantly demonstrate the effectiveness of the proposed model on diffusion prediction.
- The further ablation studies show that the generalization ability of the model is remarkably improved thanks to the effect of the network regularization.

8.1.2 Joint User Representation Learning from Diffusion and Network

- We first propose to learn behavior-shared user representations to capture strong correlations between information sharing and relationship building behaviors.
- We develop a novel joint representation learning model, which estimates latent representations of social users simultaneously from two signals, i.e., diffusion cascades and social network.
- The learned comprehensive representations are applicable for both diffusion prediction and link prediction tasks. Meanwhile, it is easy to be extended to cope with multiple behaviors.
- Compared with existing popular diffusion prediction and link prediction methods, the proposed model shows better performance on both tasks significantly.
- By capturing the interplay between diffusion and network in a latent way, the proposed model is more robust when observed cascades are insufficient or network is incomplete, which is demonstrated by the further ablation studies.

8.1.3 Sequential Neural Diffusion Model with Structure Attention

- Most previous neural diffusion models are not able to integrate network structure information. We propose a novel structure-aware sequential neural model for diffusion prediction.
- Under the recurrent neural network framework, we propose an attention mechanism to encode the structure information of diffusion users and design a gating mechanism to integrate the sequential and structural information.

- The experiments demonstrate that the proposed model remarkably improves the accuracy of diffusion predictions compared with state-of-the-art sequential neural models.

- It is also found that the structure attention contributes greatly in the proposed model. And the carefully designed gating mechanism is shown better effectiveness on information fusion than other popular gate designs.

8.1.4 Hierarchical Diffusion Attention Network

- We are the first to propose a non-RNN based neural diffusion model, i.e., hierarchical diffusion attention network (HiDAN) for diffusion prediction problem. Different from previous RNN-based models, the proposed model focuses on capturing non-sequential dependencies between users in diffusion cascades.

- The proposed model adopts two-level attention mechanisms, i.e., a dependency attention mechanism at user level to capture historical user-to-user dependencies and a influence attention at cascade level to infer possible future dependencies.

- The experiments on three real datasets demonstrate better diffusion prediction performance of the proposed model than state-of-the-art RNN-based approaches.

- The further case studies illustrate that the proposed attention mechanism is able to capture user-to-user dependencies more accurately than state-of-the-art attention-based sequential neural models.

- Thanks to the non-sequential architecture, most computations in the proposed model are parallelized as matrix computations, which are accelerated on Graph-

ics Processing Unit. Therefore, the proposed model also shows much higher efficiency than previous sequential models.

8.2 Future Work

At last, we present the following potential directions of our existing work.

- In Chapter 5, we have investigated the interplay effect between diffusion cascade and network structure under the representation learning framework. And the work has demonstrated that capturing the interplay effect is beneficial for both diffusion prediction and link prediction tasks. Meanwhile, in Chapter 7, we witnessed the state-of-the-art performance of neural networks and attention mechanisms on information diffusion prediction. Therefore, it is very interesting to ask whether we can explore the correlation between diffusion cascade and network structure by taking advantage of neural networks and attention mechanisms. Recall that the attention mechanism proposed in Chapter 7 aims at measuring the diffusion dependency strength between two users. If the correlation between diffusion and network is considered, the dependency attention should be aware of the structure closeness of the two users. On the other hand, in Chapter 6, we have explored the effectiveness of attention mechanism on capturing network structure information. If the correlation is taken into account, a new version of structure attention should be aware of the diffusion dependency strength of the two users. Therefore, for future work, we can design mutually enhanced diffusion dependency attention and structure attention mechanisms to capture the correlation between diffusion cascades and network structure in a "deeper" way.

- Another potential direction of future work is to integrate or capture more signals in the diffusion scenario. For example, the topic of the diffused message

is a very important signal. Different users may have different interests, thus the diffusion behavior vary on different topics. As for representation learning models, we can use topic label embedding or topic-specific representation space to integrate the topic information. As for neural network based models, we can develop either a conditional RNN for explicit topic information or a variational RNN for capturing latent topic. Also, we can design topic-oriented attention mechanism under the current diffusion attention network to make the model be aware of the topic. Other user-specific signals, such as user profiles or user generated text, can also be integrated in this scenario. For this kind of signals, we could extend our joint representation learning proposed in Chapter 5 to learn user embeddings from diffusion behavior, link creation behavior as well as their profile or text generation behavior. With these additional signals, we can derive more generalized and robust user representations for different prediction tasks.

- Moreover, handling dynamics of social media is also a crucial direction for future research. On the one hand, the social relationships are highly dynamics. Many network representation learning models for dynamic networks have been proposed. The key idea is incrementally updating the user representations based on network changes. We can also apply this idea under our current framework to update the representation according to the changes of network structure. On the other hand, diffusion processes are generated in real-time. Therefore, how to update the models online with newly generated diffusion cascades is also an interesting problem. Overall, how to simultaneously handle the dynamics of the two above-mentioned aspects is a challenging problem.

Bibliography

[1] Bruno Abrahao, Flavio Chierichetti, Robert Kleinberg, and Alessandro Panconesi. Trace complexity of network inference. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 491–499. ACM, 2013.

[2] Lada A Adamic and Eytan Adar. Friends and neighbors on the web. *Social networks*, 25(3):211–230, 2003.

[3] Amr Ahmed, Nino Shervashidze, Shravan Narayanamurthy, Vanja Josifovski, and Alexander J Smola. Distributed large-scale natural graph factorization. In *Proceedings of the 22nd international conference on World Wide Web*, pages 37–48. ACM, 2013.

[4] Edoardo M Airoldi, David M Blei, Stephen E Fienberg, and Eric P Xing. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9(Sep):1981–2014, 2008.

[5] Roy M Anderson, Robert M May, and B Anderson. *Infectious diseases of humans: dynamics and control*, volume 28. Wiley Online Library, 1992.

[6] Demetris Antoniades and Constantine Dovrolis. Co-evolutionary dynamics in social networks: A case study of twitter. *Computational Social Networks*, 2(1):14, 2015.

[7] Sinan Aral and Dylan Walker. Identifying influential and susceptible members of social networks. *Science*, page 1215842, 2012.

[8] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *International Conference on Learning Representations*, 2015.

[9] Norman TJ Bailey et al. *The mathematical theory of infectious diseases and its applications*. Charles Griffin & Company Ltd, 5a Crendon Street, High Wycombe, Bucks HP13 6LE., 1975.

[10] Eytan Bakshy, Jake M Hofman, Winter A Mason, and Duncan J Watts. Everyone's an influencer: quantifying influence on twitter. In *Proceedings of the*

fourth ACM international conference on Web search and data mining, pages 65–74. ACM, 2011.

[11] Roja Bandari, Sitaram Asur, and Bernardo A Huberman. The pulse of news in social media: Forecasting popularity. In *Sixth International AAAI Conference on Weblogs and Social Media*, 2012.

[12] Qing Bao, William K. Cheung, and Jiming Liu. Inferring motif-based diffusion models for social networks. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 3677–3683, 2016.

[13] Nicola Barbieri, Francesco Bonchi, and Giuseppe Manco. Topic-aware social influence propagation models. *Knowledge and information systems*, 37(3):555–584, 2013.

[14] Adrian Benton, Raman Arora, and Mark Dredze. Learning multiview embeddings of twitter users. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, volume 2, pages 14–19, 2016.

[15] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, pages 2787–2795, 2013.

[16] Christian Borgs, Michael Brautbar, Jennifer Chayes, and Brendan Lucier. Maximizing social influence in nearly optimal time. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 946–957. SIAM, 2014.

[17] Simon Bourigault, Cedric Lagnier, Sylvain Lamprier, Ludovic Denoyer, and Patrick Gallinari. Learning social network embeddings for predicting information diffusion. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, pages 393–402. ACM, 2014.

[18] Simon Bourigault, Sylvain Lamprier, and Patrick Gallinari. Representation learning for information diffusion through social networks: an embedded cascade model. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pages 573–582. ACM, 2016.

[19] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117, 1998.

[20] Francis A Buttle. Word of mouth: understanding and managing referral marketing. *Journal of strategic marketing*, 6(3):241–254, 1998.

[21] Hongyun Cai, Vincent W Zheng, and Kevin Chang. A comprehensive survey of graph embedding: problems, techniques and applications. *IEEE Transactions on Knowledge and Data Engineering*, 2018.

[22] Qi Cao, Huawei Shen, Keting Cen, Wentao Ouyang, and Xueqi Cheng. Deephawkes: Bridging the gap between prediction and understanding of information cascades. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1149–1158. ACM, 2017.

[23] Shaosheng Cao, Wei Lu, and Qiongkai Xu. Grarep: Learning graph representations with global structural information. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 891–900. ACM, 2015.

[24] Shaosheng Cao, Wei Lu, and Qiongkai Xu. Deep neural networks for learning graph representations, 2016.

[25] Shiyu Chang, Wei Han, Jiliang Tang, Guo-Jun Qi, Charu C Aggarwal, and Thomas S Huang. Heterogeneous network embedding via deep architectures. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 119–128. ACM, 2015.

[26] Jifan Chen, Qi Zhang, and Xuanjing Huang. Incorporate group information to enhance network embedding. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, pages 1901–1904, New York, NY, USA, 2016. ACM.

[27] Wei Chen, Chi Wang, and Yajun Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1029–1038. ACM, 2010.

[28] Wei Chen, Yajun Wang, and Siyu Yang. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 199–208. ACM, 2009.

[29] Justin Cheng, Lada Adamic, P Alex Dow, Jon Michael Kleinberg, and Jure Leskovec. Can cascades be predicted? In *Proceedings of the 23rd international conference on World wide web*, pages 925–936. ACM, 2014.

[30] Suqi Cheng, Huawei Shen, Junming Huang, Guoqing Zhang, and Xueqi Cheng. Staticgreedy: solving the scalability-accuracy dilemma in influence maximization. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 509–518. ACM, 2013.

[31] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, October 2014.

[32] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *ICLR*, 2016.

[33] Trevor F Cox and Michael AA Cox. *Multidimensional scaling*. CRC press, 2000.

[34] Riley Crane and Didier Sornette. Robust dynamic classes revealed by measuring the response function of a social system. *Proceedings of the National Academy of Sciences*, 105(41):15649–15653, 2008.

[35] Zhiyuan Liu Cunchao Tu, Weicheng Zhang and Maosong Sun. Max-margin deepwalk: Discriminative learning of network representation. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 3889–3895, 2016.

[36] Hadi Daneshmand, Manuel Gomez-Rodriguez, Le Song, and Bernhard Schoelkopf. Estimating diffusion network structures: Recovery conditions, sample complexity & soft-thresholding algorithm. In *ICML*, pages 793–801, 2014.

[37] Wanying Ding, Yue Shang, Lifan Guo, Xiaohua Hu, Rui Yan, and Tingting He. Video popularity prediction by sentiment propagation via implicit network. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1621–1630. ACM, 2015.

[38] Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song. Recurrent marked temporal point processes: Embedding event history to vector. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1555–1564. ACM, 2016.

[39] Nan Du, Le Song, Manuel Gomez-Rodriguez, and Hongyuan Zha. Scalable influence estimation in continuous-time diffusion networks. In *Advances in neural information processing systems*, pages 3147–3155, 2013.

[40] Nan Du, Le Song, Alexander J. Smola, and Ming Yuan. Learning networks of heterogeneous influence. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[41] Nan Du, Le Song, Hyenkyun Woo, and Hongyuan Zha. Uncover topic-sensitive information diffusion networks. In *Proceedings of the sixteenth International Conference on Artificial Intelligence and Statistics*, pages 229–237, 2013.

[42] Quang Duong, Michael P Wellman, and Satinder Singh. Modeling information diffusion in networks with unobserved links. In *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on*, pages 362–369. IEEE, 2011.

[43] Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.

[44] Mehrdad Farajtabar, Manuel Gomez-Rodriguez, Mohammad Zamani, Nan Du, Hongyuan Zha, and Le Song. Back to the past: Source identification in diffusion networks from partially observed cascades. In *AISTATS*, 2015.

[45] Mehrdad Farajtabar, Yichen Wang, Manuel Gomez-Rodriguez, Shuang Li, Hongyuan Zha, and Le Song. Coevolve: A joint point process model for information diffusion and network co-evolution. In *Advances in Neural Information Processing Systems*, pages 1954–1962, 2015.

[46] Linton C Freeman. Centrality in social networks conceptual clarification. *Social networks*, 1(3):215–239, 1978.

[47] Wojciech Galuba, Karl Aberer, Dipanjan Chakraborty, Zoran Despotovic, and Wolfgang Kellerer. Outtweeting the twitterers-predicting information cascades in microblogs. *WOSN*, 10:3–11, 2010.

[48] Shuai Gao, Jun Ma, and Zhumin Chen. Modeling and predicting retweeting dynamics on microblogging platforms. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pages 107–116. ACM, 2015.

[49] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9, pages 249–256, 2010.

[50] Jacob Goldenberg, Barak Libai, and Eitan Muller. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing letters*, 12(3):211–223, 2001.

[51] Manuel Gomez Rodriguez, Jure Leskovec, and Andreas Krause. Inferring networks of diffusion and influence. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1019–1028. ACM, 2010.

[52] Manuel Gomez Rodriguez, Jure Leskovec, and Bernhard Schölkopf. Structure and dynamics of information pathways in online media. In *Proceedings of the sixth ACM International Conference on Web Search and Data Mining*, pages 23–32. ACM, 2013.

[53] Amit Goyal, Francesco Bonchi, and Laks VS Lakshmanan. Learning influence probabilities in social networks. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 241–250. ACM, 2010.

[54] Amit Goyal, Wei Lu, and Laks VS Lakshmanan. Celf++: optimizing the greedy algorithm for influence maximization in social networks. In *Proceedings of the 20th international conference companion on World wide web*, pages 47–48. ACM, 2011.

[55] Mark Granovetter. Threshold models of collective behavior. *American journal of sociology*, pages 1420–1443, 1978.

[56] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864. ACM, 2016.

[57] Adrien Guille and Hakim Hacid. A predictive model for the temporal dynamics of information diffusion in online social networks. In *Proceedings of the 21st International Conference Companion on World Wide Web*, pages 1145–1152. ACM, 2012.

[58] Adrien Guille, Hakim Hacid, Cecile Favre, and Djamel A Zighed. Information diffusion in online social networks: A survey. *ACM SIGMOD Record*, 42(2):17–28, 2013.

[59] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and D Tikk. Session-based recommendations with recurrent neural networks. In *4th International Conference on Learning Representations, ICLR 2016*, 2016.

[60] Tuan-Anh Hoang and Ee-Peng Lim. Virality and susceptibility in information diffusions. In *ICWSM*, 2012.

[61] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[62] Qingbo Hu, Sihong Xie, Shuyang Lin, Senzhang Wang, and Philip S. Yu. CENI: A hybrid framework for efficiently inferring information networks. In *Proceedings of the Ninth International Conference on Web and Social Media, ICWSM 2015, University of Oxford, Oxford, UK, May 26-29, 2015*, pages 618–621, 2015.

[63] Zepeng Huo, Xiao Huang, and Xia Hu. Link prediction with personalized social influence. In *AAAI*, 2018.

[64] Tomoharu Iwata, Amar Shah, and Zoubin Ghahramani. Discovering latent influence in online social activities via shared cascade poisson processes. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 266–274. ACM, 2013.

[65] Glen Jeh and Jennifer Widom. Simrank: a measure of structural-context similarity. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 538–543. ACM, 2002.

[66] Leo Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, 1953.

[67] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. ACM, 2003.

[68] Jinha Kim, Wonyeol Lee, and Hwanjo Yu. Ct-ic: Continuously activated and time-restricted independent cascade model for viral marketing. *Knowledge-Based Systems*, 62:57–68, 2014.

[69] Masahiro Kimura and Kazumi Saito. Tractable models for information diffusion in social networks. In *European conference on principles of data mining and knowledge discovery*, pages 259–271. Springer, 2006.

[70] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[71] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[72] Thomas N Kipf and Max Welling. Variational graph auto-encoders. *NIPS Workshop on Bayesian Deep Learning*, 2016.

[73] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, (8):30–37, 2009.

[74] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[75] Chris Kuhlman, V Kumar, Madhav Marathe, S Ravi, D Rosenkrantz, Samarth Swarup, and Gaurav Tuli. A bi-threshold model of complex contagion and its application to the spread of smoking behavior. In *Proceedings of the workshop on social network mining and analysis (SNA-KDD 2011)*, 2011.

[76] Takeshi Kurashima, Tomoharu Iwata, Noriko Takaya, and Hiroshi Sawada. Probabilistic latent network visualization: inferring and embedding diffusion networks. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1236–1245. ACM, 2014.

[77] John Lafferty and Guy Lebanon. Diffusion kernels on statistical manifolds. *Journal of Machine Learning Research*, 6(Jan):129–163, 2005.

[78] Cédric Lagnier, Ludovic Denoyer, Eric Gaussier, and Patrick Gallinari. Predicting information diffusion in social networks using content and user's profiles. In *European Conference on Information Retrieval*, pages 74–85. Springer, 2013.

[79] Theodoros Lappas, Evimaria Terzi, Dimitrios Gunopulos, and Heikki Mannila. Finding effectors in social networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1059–1068. ACM, 2010.

[80] Jerald F Lawless. *Statistical models and methods for lifetime data*, volume 362. John Wiley & Sons, 2011.

[81] Elisa T Lee and John Wang. *Statistical methods for survival data analysis*, volume 476. John Wiley & Sons, 2003.

[82] Jure Leskovec, Lars Backstrom, and Jon Kleinberg. Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 497–506. ACM, 2009.

[83] Jure Leskovec, Deepayan Chakrabarti, Jon Kleinberg, Christos Faloutsos, and Zoubin Ghahramani. Kronecker graphs: An approach to modeling networks. *Journal of Machine Learning Research*, 11(Feb):985–1042, 2010.

[84] Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 420–429. ACM, 2007.

[85] Jure Leskovec, Mary McGlohon, Christos Faloutsos, Natalie S Glance, and Matthew Hurst. Patterns of cascading behavior in large blog graphs. In *SDM*, volume 7, pages 551–556, 2007.

[86] Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems*, pages 2177–2185, 2014.

[87] Cheng Li, Jiaqi Ma, Xiaoxiao Guo, and Qiaozhu Mei. Deepcas: An end-to-end predictor of information cascades. In *Proceedings of the 26th International Conference on World Wide Web*, pages 577–586. International World Wide Web Conferences Steering Committee, 2017.

[88] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1419–1428. ACM, 2017.

[89] Jiwei Li, Alan Ritter, and Dan Jurafsky. Learning multi-faceted representations of individuals from heterogeneous evidence using neural networks. *CoRR*, abs/1510.05198, 2015.

[90] Juzheng Li, Jun Zhu, and Bo Zhang. Discriminative deep random walk for network classification. *Proceedings of ACL-12*, 2016.

[91] David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7):1019–1031, 2007.

[92] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. *International Conference on Learning Representations*, 2017.

[93] Pengfei Liu, Shafiq Joty, and Helen Meng. Fine-grained opinion mining with recurrent neural networks and word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1433–1443, 2015.

[94] Weiping Liu and Linyuan Lü. Link prediction based on local random walk. *EPL (Europhysics Letters)*, 89(5):58007, 2010.

[95] Linyuan Lü and Tao Zhou. Link prediction in complex networks: A survey. *Physica A: statistical mechanics and its applications*, 390(6):1150–1170, 2011.

[96] Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, 2015.

[97] Eren Manavoglu, Dmitry Pavlov, and C Lee Giles. Probabilistic user behavior models. In *Third IEEE International Conference on Data Mining*, pages 203–210. IEEE, 2003.

[98] Víctor Martínez, Fernando Berzal, and Juan-Carlos Cubero. A survey of link prediction in complex networks. *ACM Computing Surveys (CSUR)*, 49(4):69, 2017.

[99] Aditya Krishna Menon and Charles Elkan. Link prediction via matrix factorization. In *Joint european conference on machine learning and knowledge discovery in databases*, pages 437–452. Springer, 2011.

[100] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[101] Swapnil Mishra, Marian-Andrei Rizoiu, and Lexing Xie. Feature driven and point process approaches for popularity prediction. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 1069–1078. ACM, 2016.

[102] Seth Myers and Jure Leskovec. On the convexity of latent social network inference. In *Advances in Neural Information Processing Systems*, pages 1741–1749, 2010.

[103] Annamalai Narayanan, Mahinthan Chandramohan, Lihui Chen, Yang Liu, and Santhoshkumar Saminathan. subgraph2vec: Learning distributed representations of rooted sub-graphs from large graphs. *CoRR*, abs/1606.08928, 2016.

[104] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33, 2016.

[105] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. *arXiv preprint arXiv:1605.05273*, 2016.

[106] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[107] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 701–710. ACM, 2014.

[108] Sasa Petrovic, Miles Osborne, and Victor Lavrenko. Rt to win! predicting message propagation in twitter. In *ICWSM*, 2011.

[109] B Aditya Prakash, Jilles Vreeken, and Christos Faloutsos. Spotting culprits in epidemics: How many and which ones? In *2012 IEEE 12th International Conference on Data Mining*, pages 11–20. IEEE, 2012.

[110] Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 459–467. ACM, 2018.

[111] Manuel G Rodriguez and Bernhard Schölkopf. Submodular inference of diffusion networks from multiple trees. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 489–496, 2012.

[112] Manuel Gomez Rodriguez, David Balduzzi, and Bernhard Schölkopf. Uncovering the temporal dynamics of diffusion networks. In Lise Getoor and Tobias Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, ICML '11, pages 561–568, New York, NY, USA, June 2011. ACM.

[113] Yu Rong, Qiankun Zhu, and Hong Cheng. A model-free approach to infer the diffusion network from event cascade. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, pages 1653–1662, New York, NY, USA, 2016. ACM.

[114] Kazumi Saito, Masahiro Kimura, Kouzou Ohara, and Hiroshi Motoda. Learning continuous-time information diffusion model for social behavioral data analysis. In *Asian Conference on Machine Learning*, pages 322–337. Springer, 2009.

[115] Kazumi Saito, Masahiro Kimura, Kouzou Ohara, and Hiroshi Motoda. Behavioral analyses of information diffusion models by observed data of social network. In *International conference on social computing, behavioral modeling, and prediction*, pages 149–158. Springer, 2010.

[116] Kazumi Saito, Masahiro Kimura, Kouzou Ohara, and Hiroshi Motoda. Efficient estimation of cumulative influence for multiple activation information diffusion model with continuous time delay. In *Pacific Rim International Conference on Artificial Intelligence*, pages 244–255. Springer, 2010.

[117] Kazumi Saito, Masahiro Kimura, Kouzou Ohara, and Hiroshi Motoda. Generative models of information diffusion with asynchronous timedelay. In *Asian Conference on Machine Learning*, pages 193–208. Springer, 2010.

[118] Kazumi Saito, Ryohei Nakano, and Masahiro Kimura. Prediction of information diffusion probabilities for independent cascade model. In *Knowledge-based Intelligent Information and Engineering Systems*, pages 67–75. Springer, 2008.

[119] Kazumi Saito, Kouzou Ohara, Yuki Yamagishi, Masahiro Kimura, and Hiroshi Motoda. Learning diffusion probability based on node attributes in social networks. In *International Symposium on Methodologies for Intelligent Systems*, pages 153–162. Springer, 2011.

[120] Gerard Salton and Michael J McGill. Introduction to modern information retrieval. 1986.

[121] John Scott. *Social network analysis*. Sage, 2012.

[122] Hua-Wei Shen, Dashun Wang, Chaoming Song, and Albert-László Barabási. Modeling and predicting popularity dynamics via reinforced poisson processes. *arXiv preprint arXiv:1401.0778*, 2014.

[123] Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan, and Chengqi Zhang. Disan: Directional self-attention network for rnn/cnn-free language understanding. In *AAAI Conference on Artificial Intelligence*, 2018.

[124] Nino Shervashidze, Pascal Schweitzer, Erik Jan van Leeuwen, Kurt Mehlhorn, and Karsten M Borgwardt. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12(Sep):2539–2561, 2011.

[125] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958, 2014.

[126] Duyu Tang, Bing Qin, and Ting Liu. Document modeling with gated recurrent neural network for sentiment classification. In *EMNLP*, pages 1422–1432, 2015.

[127] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1067–1077. International World Wide Web Conferences Steering Committee, 2015.

[128] Lei Tang and Huan Liu. Scalable learning of collective behavior based on sparse social dimensions. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1107–1116. ACM, 2009.

[129] Youze Tang, Xiaokui Xiao, and Yanchen Shi. Influence maximization: Near-optimal time complexity meets practical efficiency. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 75–86. ACM, 2014.

[130] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

[131] Cunchao Tu, Hao Wang, Xiangkai Zeng, Zhiyuan Liu, and Maosong Sun. Community-enhanced network representation learning for network analysis. *arXiv preprint arXiv:1611.06645*, 2016.

[132] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010. 2017.

[133] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. *International Conference on Learning Representations*, 2018.

[134] Greg Ver Steeg and Aram Galstyan. Information-theoretic measures of influence based on content dynamics. In *Proceedings of the sixth ACM International Conference on Web Search and Data Mining*, pages 3–12. ACM, 2013.

[135] Daixin Wang, Peng Cui, and Wenwu Zhu. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1225–1234, 2016.

[136] Jia Wang, Vincent W Zheng, Zemin Liu, and Kevin Chen-Chuan Chang. Topological recurrent neural network for diffusion prediction. In *Data Mining (ICDM), 2017 IEEE International Conference on*, pages 475–484. IEEE, 2017.

[137] Senzhang Wang, Xia Hu, Philip S Yu, and Zhoujun Li. Mmrate: inferring multi-aspect diffusion networks with multi-pattern cascades. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1246–1255. ACM, 2014.

[138] Yongqing Wang, Huawei Shen, Shenghua Liu, and Xueqi Cheng. Learning user-specific latent influence and susceptibility from information cascades. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 477–483, 2015.

[139] Yongqing Wang, Huawei Shen, Shenghua Liu, Jinhua Gao, and Xueqi Cheng. Cascade dynamics modeling with attention-based recurrent neural network. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pages 2985–2991, 2017.

[140] Yu Wang, Gao Cong, Guojie Song, and Kunqing Xie. Community-based greedy algorithm for mining top-k influential nodes in mobile social networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1039–1048. ACM, 2010.

[141] Zhitao Wang, Chengyao Chen, and Wenjie Li. Predictive network representation learning for link prediction. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 969–972. ACM, 2017.

[142] Zhitao Wang, Chengyao Chen, and Wenjie Li. Attention network for information diffusion prediction. In *Companion Proceedings of the The Web Conference 2018*, pages 65–66, 2018.

[143] Zhitao Wang, Chengyao Chen, and Wenjie Li. A sequential neural information diffusion model with structure attention. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1795–1798. ACM, 2018.

[144] Zhitao Wang, Chengyao Chen, and Wenjie Li. Information diffusion prediction with network regularized role-based user representation learning. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 13(3):29, 2019.

[145] Zhitao Wang, Chengyao Chen, Ke Zhang, Yu Lei, and Wenjie Li. Variational recurrent model for session-based recommendation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1839–1842, 2018.

[146] Zhitao Wang, Yu Lei, and Wenjie Li. Neighborhood interaction attention network for link prediction. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2153–2156, 2019.

[147] Zhitao Wang and Wenjie Li. Hierarchical diffusion attention network. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 3828–3834. International Joint Conferences on Artificial Intelligence Organization, 7 2019.

[148] L. Weng, F Menczer, and Y. Y. Ahn. Virality prediction and community structure in social networks. *Scientific Reports*, 3(8):618–618, 2013.

[149] Lilian Weng, Jacob Ratkiewicz, Nicola Perra, Bruno Gonçalves, Carlos Castillo, Francesco Bonchi, Rossano Schifanella, Filippo Menczer, and Alessandro Flammini. The role of information diffusion in the evolution of social networks. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 356–364. ACM, 2013.

[150] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015.

[151] Pinar Yanardag and SVN Vishwanathan. Deep graph kernels. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1365–1374. ACM, 2015.

[152] Cheng Yang and Zhiyuan Liu. Comprehend deepwalk as matrix factorization. *arXiv preprint arXiv:1501.00358*, 2015.

[153] Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Y Chang. Network representation learning with rich text information. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina*, pages 2111–2117, 2015.

[154] Jaewon Yang and Jure Leskovec. Modeling information diffusion in implicit networks. In *2010 IEEE International Conference on Data Mining*, pages 599–608. IEEE, 2010.

[155] Shuang-Hong Yang and Hongyuan Zha. Mixture of mutually exciting processes for viral diffusion. *ICML (2)*, 28:1–9, 2013.

[156] Yang Yang, Jie Tang, Cane Wing-ki Leung, Yizhou Sun, Qicong Chen, Juanzi Li, and Qiang Yang. Rain: Social role-aware information diffusion. In *AAAI*, pages 367–373, 2015.

[157] Hongyi Zhang, Tong Zhao, Irwin King, and Michael R. Lyu. Modeling the homophily effect between links and communities for overlapping community detection. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 3938–3944, 2016.

[158] Jing Zhang, Biao Liu, Jie Tang, Ting Chen, and Juanzi Li. Social influence locality for modeling retweeting behaviors. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China*, volume 13, pages 2761–2767, 2013.

[159] Muhan Zhang and Yixin Chen. Weisfeiler-lehman neural machine for link prediction. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 575–583. ACM, 2017.

[160] Qingyuan Zhao, Murat A Erdogdu, Hera Y He, Anand Rajaraman, and Jure Leskovec. Seismic: A self-exciting point process model for predicting tweet popularity. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1513–1522. ACM, 2015.

[161] Zhou Zhao, Ben Gao, Vincent W Zheng, Deng Cai, Xiaofei He, and Yueting Zhuang. Link prediction via ranking metric dual-level attention network learning. In *IJCAI*, pages 3525–3531, 2017.

[162] Vincent W Zheng, Sandro Cavallari, Hongyun Cai, Kevin Chen-Chuan Chang, and Erik Cambria. From node embedding to community embedding. *arXiv preprint arXiv:1610.09950*, 2016.

[163] Chang Zhou, Yuqiong Liu, Xiaofei Liu, Zhongyi Liu, and Jun Gao. Scalable graph embedding for asymmetric proximity. In *AAAI*, pages 2942–2948, 2017.

[164] Chuan Zhou, Peng Zhang, Wenyu Zang, and Li Guo. On the upper bounds of spread for greedy algorithms in social network influence maximization. *IEEE Transactions on Knowledge and Data Engineering*, 27(10):2770–2783, 2015.

[165] Tao Zhou, Linyuan Lü, and Yi-Cheng Zhang. Predicting missing links via local information. *The European Physical Journal B*, 71(4):623–630, 2009.