USER PROFILING IN SOCIAL NETWORKS BASED ON GRAPH DATA: BRIDGING LOCAL AND GLOBAL STRUCTURES

BY

AMIN JAVARI

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois Urbana-Champaign, 2021

Urbana, Illinois

Doctoral Committee:

Professor Kevin Chenchuan Chang, Chair
Professor Cheng Xiang Zhai
Associate Professor Hari Sundaram
Assistant Professor Jiliang Tang, Michigan State University

ABSTRACT

User profiling plays a key role in adaptive systems on online social networks (OSN). Building user profiles allows to identify user's interests (interest modeling) and predict their future behaviors (behavioral modeling). While various data sources can be exploited on OSNs for user profiling, graph data is a more structured and more abundant data source compared to other data sources like content data. In fact, graphs can be used to capture different types of interactions and relationships among users. By exploiting graph data, users can be profiled based on their *neighbors*.

In this thesis, we aim to address different profiling problems based on graph data, including interest modeling and behavioral modeling, where the main focus is on integrating local and global structures. In fact, to successfully exploit graph data for user profiling, both local and global structures need to be analyzed, i.e., profiling a node only based on its immediate neighbors (local structures) is not fully effective. Such models miss the precious information that can be exploited by analyzing the entire network (global structures). However, as the network can be large, and the connection between nodes may be complex, building *efficient* and *effective* models that can bridge local and global structures is challenging.

We define novel interest profiling problems on interest-based networks that require us to bridge local and global structures *efficiently*. In particular, we focus on followee/follower networks in microblogging websites as interest-based networks. To address the efficiency challenge of interest profiling, we propose the notion of profiling based on hub nodes. The hub nodes can be viewed as the cluster centers of the network where each cluster depicts a certain interest topic. Our core idea is that users can efficiently be profiled by transitively profiling them based on the hub nodes they are connected to.

We also investigate user behavior modeling on networks that represent user's behavioral interactions. Various methods have been introduced for user modeling on classic networks with a single type of interaction that are able to bridge local and global structures. However, their effectiveness is limited on networks with multi-types of interactions due to the *complexity* of relations on such networks. We introduce models to build behavioral models in networks with multi-types of interactions that systematically address the complexity challenge.

To my family for all their love.

ACKNOWLEDGMENTS

I am deeply grateful to my advisor, Professor Kevin Chang, for his patience, unfailing help, and kind support throughout my Ph.D. I was truly blessed to have Kevin as my advisor. His keen insights, his bright ideas and intuitions, and his rigorous coaching helped me broaden my vision, think more critically, and dig deeper in my research.

I would like to express my sincere gratitude to my thesis committee members, Professor ChengXiang Zhai, Professor Hari Sundaram, and Professor Jiliang Tang, for offering me their thoughtful comments and their detailed suggestions, which helped me to immensely improve this thesis.

I would also like to thank my close collaborators during my Ph.D. years, including Pouya Esmailian, Zijie Huang, Zhankui He, Tyler Derr, HongXiang Qiu, and Mehrab Norouzitallab, who have broadened and deepened my research. This thesis would not be possible without their helps.

I want to thank my lab mates and the Data and Information Systems Laboratory (DAIS) members for their valuable suggestions and stimulating discussions that I had over the years. To name a few: Mangesh Bender, Long Pham, Aravind Sankar, Himel Dev, Hongtai Cao, and Amirhossein Aleyasen.

Last, but not least, my deepest gratitude and appreciation belongs to my parents and my brother for their encouragement, selfless support, and unconditional love throughout my life. This thesis is dedicated to them.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

Online platforms and social networking (OSNs) are precious data sources for user profiling. On these websites, billions of users connect to and communicate/interact with other users, share their opinions, and demonstrate their interests in different ways, which can be used to profile them. The broad terms of user modeling/profiling may refer to different meanings. In the context of adaptive systems, Brusilovsky et al. define a user model as, "a representation of information about an individual user that is essential for an adaptive system to provide the adaptation effect" [1]. With this definition, in a general setting, the explainability/interpretability of user-profiles is not a key objective for the profiling process. That is, any form of user representation that can aid us to provide penalization/adaptation in a system is considered a user model. However, building interpretable profiles, e.g., representing users based on a set of keywords, in some certain tasks like recommendation with explanation is of great importance.

From the application perspective, user profiling can be categorized into interest modeling and behavioral/intention modeling [2]. In interest modeling, the goal is to identify user's interests and preferences based on her previous interactions, while in behavior modeling, the task is to build a model of user's previous interactions to predict their future actions or to estimate the intention of their actions. In both cases, user modeling plays a key role in providing personalized services on OSNs and third-party applications [3].

It should be noted from a mathematical point of view, both classes of user modeling can be formulated as a preference/matching prediction problem. Having a network of user's connections and given a user u and an entity t, the goal is to determine the label of the interaction between u and t. For example, in a user-to-user interaction prediction problem, t is a user within the target graph. Other example is item recommendation task where t is an item from a target pool of items used for recommendation. With this formulation, even building interpretable user profiles can be regarded as determining the relevance of a user to a set of interest categories represented by some keywords.

Various data sources can be used to build user profiles for different purposes in OSNs [3]. In general, the actions of a user in an OSN can be mainly captured using content data and graph data. Take Twitter as an example where a user may generate content data in different ways, e.g., tweeting, re-tweeting, liking, replying to a tweet, and describing themselves in their profiles [4]. The content data available on OSNs is a valuable data source for user profiling. Apart from the content data, the graph data can be used to model different types of relationships and interactions between users, which can be employed for user modeling

[5, 6].

In this thesis, we mainly focus on the challenges of user profiling based on graph data. In fact, user modeling based on content data faces some fundamental issues. One of the major issues is that it is often short and noisy. Moreover, user modeling based on content data requires users to be active, which is not always the case. In fact, it has been shown that a large proportion of Twitter users are passive ones who do not generate any content [7]. However, unlike content data, graph data is a more structured and more abundant data source compared to content data [4]. For example, on Twitter, a user's followee/follower relationships can be captured as graph data, and even silent/listening users tend to create such relationships. Hence, leveraging information from the social networks of a user can be helpful for user profiling, especially for passive users.

While our focus is on graph data, to build a comprehensive user profiling system, all data sources associated with a user should be taken into account. In fact, the knowledge extracted from different data sources can be viewed as signals that allow us to model a user from different perspectives. Hence, these informative signals can be fused to build a unified model of the target users. For example, users' tweets along with their connections can be leveraged to predict their future interactions and identify their interests. However, we mainly investigate graph data for user profiling because it allows us to dedicatedly understand and approach the challenges associated with graph data. Also, different works have shown that a model based on graph data can be combined with content-based models [8, 9, 10]. Hence, it can be said that a model developed based on graph data can be extended to accommodate other data sources.

1.1 MOTIVATION

When building graph-based user profiling models a key consideration is to exploit both **local** and **global** structures of the input graph in an efficient and effective manner. Indeed, user modeling based on graph data allows us to profile users based on the profiles of their 'neighbors', i.e., having the interest/behavioral models of a user's neighbors, a model of the target user can be constructed transitively. Generally speaking, local structures can be regarded as a sub-graph that consists of the target users and it's 'close' neighbors, e.g., immediate neighbors, while global structures can be viewed as connectivity patterns of the target user towards higher-order neighbors and also cluster-based structures.

More formally, we define global structures of a graph as those structures that can be obtained only by analyzing the entire network or at least a large proportion of the network. For example, clusters of a network as a representative of global structures cannot be obtained

by examining the individual nodes in a given graph. On the other hand, we define local structures as individual nodes or their network neighbors. Based on these definitions, higher-order neighbors (e.g. two-hop neighbors) of nodes can be regarded as structures that go beyond immediate neighbors and aim to capture some levels of global structures. In fact, it can be said that as we move from immediate neighbors of a node towards its higher-order neighbors, we make a transition in a spectrum from local structures towards global structures.

A local-structure-based model, which attempts to profile a node based on it's local neighbors, may encounter significant drawbacks. Most of the real-world networks are sparse networks, which means that local neighbors with known profiles may not exist to profile the target user [11, 12]. Hence, such models are vulnerable to the sparsity issue. Moreover, even in the presence of a significant number of local neighbors, still valuable information can be exploited from global structures, i.e., the graph clusters and higher-order neighbors of the target user can be taken into account to build a more accurate/reliable profile of the user.

Although global structures are quite informative, exploiting such structures is challenging. Doing so may affect the *efficiency* of the user profiling process, i.e., analyzing the entire network to profile users may have burdensome computations which is not ideal for networks with high dynamics or application that require online and real time profiling of users. Moreover, involving higher-order neighbors in the profiling process may become very challenging due to the *complexity* of connectivity patterns and relations among neighbors in some networks. Let's assume in a network where connections may carry different meanings, both users m and n are two-hop neighbors of the user u. Are they equally important in the profiling of u? Do we need to involve them in the profiling process of u in the same manner?

1.2 CONTRIBUTIONS AND OUTLINE

In this thesis, we aim to address different graph-based profiling problems, including interest modeling and behavioral modeling on different network types, where the focus is on integrating local and global structures. In fact, the reason behind defining our problems on both interaction-based and interest-based networks is that these two types of networks pose different structures, and user profiling on them may encounter different challenges. **Chapter 2** introduces the overview of techniques introduced for bridging local and global structures, and studies them from two perspectives: efficiency and complexity.

In **Chapters 3 and 4**, we define user profiling problems in interest-based networks that require us to efficiently exploit both global and local structures of the target network to build user profiles. More specifically, we define our target problems on microblogging networks

as representatives of interest-based networks. On microblogging websites, a large portion of followee connections represent interest topics, i.e., when a user becomes interested in a topic follows users associated with that topic [5]. However, as mentioned, the existing models do not effectively address the efficiency challenge on such networks. In fact, models based on exploiting longer cycles and those based on clustering are not ideal in terms of the efficiency challenge [13, 14]. In order to approach the efficiency challenge, as a general idea, we suggest that the approach to bridge the local and global structures can leverage the topology of the network. Interest-based graphs like microblogging networks in which connections directly/explicitly indicate user's interests tend to have scale-free structures. Our idea is that the scale-freeness of such networks is quite beneficial in building *efficient* user modeling techniques.

Additionally, in **Chapters 5 and 6**, we investigate user profiling problems on interaction-based networks. It can be said that the existing methods for user modeling on classic interaction-based have delivered promising results. In particular, Graph Neural Network (GNN) based models can be considered as successful approaches for user modeling that potentially can bridge local and global structures [15]. However, the effectiveness of such models is limited in more complex settings, i.e., when the interactions between user have multiple types. In this thesis, we mainly focus on signed networks as a representative of interaction-based networks with multi-type of interactions where the connections between nodes have two types of links [16]. We introduce models for user modeling in these networks that allow to systematically bridge local and global structures. It is worth noting while the proposed have been applied to signed networks, they can be easily applied to networks where connections have more than two types. In the following, the problems associated with each network type are introduced.

1.2.1 Interest-Based Networks

Before introducing the problems defined on microblogging networks, we briefly introduce the structure of microblogging networks because the main idea of our user profiling models takes advantage of their structural properties.

Scale-freeness of microbloggin networks. Microblogging websites like Twitter and Weibo can be considered as scale-free networks [7, 17, 18]. The scale-freeness of these networks stems from the link generation process of these networks. In addition, it has been shown in previous works that the link generation process of individual nodes can determine the global structure of a network [19]. An important class of networks where the link generation process within the local structure decides the global structures is scale-free

networks. Generally speaking, if the nodes in a network are more likely to make connections towards nodes with higher popularity (indegree), it can lead to the higher scale-freeness of the networks [19]. This notion of preferential attachment is heavily involved in the link generation process of microblogging networks [5, 7].

In fact, there are two main reasons for the creation of a link from an initiator node to a receiver node in a microblogging website. First, the receiver/followee node represents a topic interesting to the initiator/follower node [5, 7]. And, indeed, most of the links are of this type. Second, a social bond exists between the receiver and initiator users. A limited number of social tie-based connections can be received/created by a user. Hence, it can be concluded when a node becomes popular in a microblogging network, it is a major representative of a topic of interest (or a set of interests). In fact, there exist some popular representative nodes for each topic of interest, and if a user is interested in a topic, she very likely follows some of the corresponding representative nodes. Based on this generative process of links in microblogging websites, we observe that microblogging networks are scale-free networks, i.e., they contain some hub nodes, and a large portion of links are toward hub nodes [18].

Integrating local and global structures efficiently. We propose that hub nodes in interest-based scale-free networks can be viewed as precious elements of the networks, which enable us to efficiently and systematically bridge/integrate local and global structures. Hub nodes lower the network diameter, and the distance between nodes with shared interests/attributes, i.e., for each cluster of users sharing the same interest, *interest-based clusters*, there exist some hub nodes and most of the connections of cluster members are towards the hub nodes. As such, hub node's profiles describe their corresponding clusters.

We believe while hub nodes are within local structures of nodes, they also represent global structures, i.e., the skeleton of a network can be obtained by only profiling hub nodes as they represent the interest-based cluster centers. From this perspective, modeling hub-nodes can be viewed as a method for clustering networks. While these cluster centers are immediate neighbors of users, they effectively represent global structures. As such, our key idea is that given an interest-based scale-free network, by profiling a very small portion of the nodes, hub nodes, we can identify the graph clusters, profile clusters and transitively derive the profiles of other users. While this method is quite efficient, it leverages both local and global structures. We introduce different social-network analysis models by relying on this general idea.

Chapter 3: Graph-based hashtag recommendation. Interest modeling can be defined as building a function $pref(u, i)$, which represents how much user u likes/dislikes item i where the function can be obtained by analyzing different data sources associated with u [20]. This broad definition can cover various applications of interest-based personalization

in adaptive information systems and search engines in different frameworks, including social networking and social media, e-commerce, security, and access control. A classic and well-studied application of personalizing that complies with this definition is the recommendation problem. In fact, the majority of the previous studies on interest profiling have been conducted with the purpose of predicting user interests followed by recommending different types of content, such as news, URLs, publications, and tweets [3].

Following this line of research, we apply our idea for graph-based interest profiling to the hashtag recommendation problem in microblogging networks, i.e., given the relevance of a set of user's to a set of hashtags (input labeled set of users), the task is to determine the relevance of other users in the network to hashtags based on analyzing the network structure. It should be mentioned, although we focus on the hashtag recommendation problem, it can be adopted to address different problems associated with personalization. In fact, in this family of problems, given a labeled set of users (here the labels are hashtags associated with each user), the task becomes propagating 'labels' in the network. Towards solving the problem, recently, a large body of research has been conducted, and a class of models denoted as Graph Convolutional Networks (GCN) [15] have been introduced. However, the effectiveness of such models in certain settings/applications is limited. In fact, the dynamics of the network, efficiency of the profiling model, and also limitedness of labeled users are important factors in designing profiling models that are not fully considered in the existing models.

To address these challenges, we leverage the concept of representative nodes [21]. Given a training set of users and having the labels of their relevance to hashtags, we transform hub nodes and hashtags into a shared latent space. To predict the relevance score of a user to a hashtag, we aggregate the embeddings of the hub nodes followed by the user in a weighted way (where the weights depend on the target hashtag) and finally determine the relevance of the obtained embedding vector of the users to that of the target hashtag. We show that this structure can easily leverage both local and global structures, handle the dynamics of the network, and is quite efficient. Moreover, unlike classic node embedding models that generate fixed embeddings, our proposed model builds target-dependent embeddings of users with respect to a given hashtag, which improves the quality of recommendations.

Chapter 4: Graph-based user discovery. Along with the classic task of recommendation, we also introduce/investigate the problem of interest-based user discovery on microblogging websites: given a set of users with shared interest as the input query, the task is to retrieve other users in the network with similar interests. This problem setting is different from classic graph-based profiling problems. Unlike a classic problem where the labels for the input set are given [15], in the proposed problem, an input seed set is not labeled.

Hence, to approach the problem, both tasks of 'labeling' and 'propagation of labels' should be addressed. In other words, we need to identify the linkage patterns that represent shared interest patterns and next rank other users in the network with respect to the similarity of their linkage patterns to those extracted from the seed users.

We address the problem again by relying on the notion of representative/hub nodes. We show that the notion of representative nodes not only helps to model the interests of users but also it aids in indexing and discovering users with shared interests. Based on this idea, rather than analyzing the entire network, we analyze the connections of seed users towards hub nodes and determine the groups of hub nodes representing each of the topics embedded in the input seed set. To this end, we propose a carefully designed graphical model. Next, we discover and rank candidate users according to the similarity of their linkage patterns towards hub nodes to those patterns extracted from seed users. In summary, in this model, we exploit representative/hub nodes to label seed users, discover candidate users, and ultimately rank candidate users in an efficient way.

1.2.2 Interaction-Based Networks

As discussed, the interaction between users can also be modeled as graph data. Various applications can be considered for behavioral modeling based on the history of user's interactions. In particular, behavioral modeling is important for predicting user's future interactions, which is beneficial for tasks like determining the effectiveness of promotional campaigns, cross-marketing strategies, and anomaly detection [2]. Also, it is worth noting that behavior modeling can be employed to address the task of interest modeling in an indirect way [6]. Although most interactions between users do not explicitly indicate a certain shared interest between users, according to the notion of homophily, such interactions may imply shared interests between them. For example, having a set of users holding a certain opinion/interest, analyzing their interactions can aid in identifying like-minded people, i.e., we can analyze how tightly two nodes are connected in the network to determine how possibly they may share the same opinions/interests [6].

Signed networks as a representative of networks with multi-types of interactions. The interactions between users may have different types [22]. A representative of the networks with multi-types of links/interactions is signed networks [16]. In signed networks, the interactions between users may have both positive and negative meanings. For example, a variety of interactions on social media sites, such as Amazon, Wikipedia, and Epinions, can be represented as a signed network where positive signs represent trust, agreement, or friendship, while negative ones may show distrust, disagreement, or enmity [23]. The underlying

principles of signed networks can be substantially different from those of classic networks due to the existence of both positive and negative links. We study two problems on signed networks: link label prediction (sign prediction) and representation learning (which can be used for link prediction). These two tasks can be regarded as behavioral modeling problems, i.e., by analyzing the previous of users, we aim to predict their future interactions, which has applications in trust prediction and anomaly detection [24]. Moreover, these tasks may be counted as tasks associated with interest profiling [24]. For example, addressing sign/link prediction in signed networks can be used to build a user to user recommender system in social networks. Analogous to interest-based networks, it is important to capture both local and global structures in signed networks to address the outlined tasks [11]. However, the complexity of paths connecting users/nodes makes leveraging global structures quite challenging.

Chapter 5: Probabilistic sign prediction. The sign prediction task is defined as predicting the sign of the edge between two nodes, which can be viewed as a behavioral modeling task, i.e., the model can be used to predict the labels of the interactions of a given user with other users. The problem has been extensively studied in the literature [24]. Some of the methods introduced for the sign prediction problem solely rely on local information to predict the sign of the edge in question [25]. Another class of the predictors seek to use only global structures to address the sparsity problem [26]. In all, previously introduced models do not fully leverage the rich information contained in both local and global structures.

In this work, we introduce a probabilistic model for sign prediction that relies on a set of parameters where those parameters can be estimated based on both local and global structures [27]. We suggest that local information specifically describes what is happening between target nodes, but the information obtained from local structures may end up being unreliable due to the data sparseness. One the other hand, global structures tend to provide reliable information; however, naturally, such structures could not capture detailed information about target links. We solve the dilemma using the idea of smoothing, i.e., to estimate the parameters of our probabilistic model, we smooth the estimations obtained from local structures with those obtained from global structures as background knowledge, where the smoothing weights are defined according to the statistical reliability of local structures.

Chapter 6: Node embedding in signed networks. Representation learning over signed networks not only allows us to capture link labels but also enables us to embed link structures. Hence, it can be employed to address various tasks associated with behavioral modeling, including sign prediction, link prediction, and network clustering [28]. Embedding signed network has recently gained lots of attention in the literature [29, 30, 31, 32]. Analogous to a large class of embedding models in unsigned networks, these models try to

embed the network by finding the similarities between nodes based on the paths connecting them. In unsigned networks, a path between two nodes represents their closeness [28]. While employing path-based similarities is an effective strategy to integrate both local and global structures, defining path-based similarities in signed networks is challenging since a signed path may indicate either closeness or distantness. That is why most of the works rely on paths with a length one or two to find node similarities, which obviously has limited effectiveness in capturing global structures [30, 33]. Recent works have attempted to involve higher-order paths by relying on some strong assumptions about the origin of the network, which consequently injects noise into the embedding process [29, 34].

To approach the challenge of the complexity of paths, we introduce a novel framework able to capture higher-order proximities without making any assumptions [35]. We lay out a new perspective for network embedding denoted as network transformation-based embedding: if embedding a network is challenging, it can be transformed into another network for which the embedding task has lower complexity. The transformation can be done by mapping each node in the original network to multiple nodes in the transformed network. Next, the transformed network can be embedded. Finally, the embedding vectors obtained from the transformed network can be aggregated to encode the original network. Based on this general idea, we transform the input signed network to an unsigned network, embed the unsigned network by adopting the classic embedding models, and eventually integrate the obtained embeddings to encode the original signed network.

CHAPTER 2: BACKGROUND

The idea of integrating local and global structures has been investigated in different ways for different user profiling problems [13, 36, 37, 38, 39]. These techniques range from classic feature-based link prediction models to recent graph convolutional network-based models. However, the strategies employed in these works share similar principles. In fact, two important classes can be considered for leveraging both local and global structures: (1) models based on higher-cycle paths between nodes, and (2) models based on cluster structures.

A major class of the techniques try to go beyond immediate neighbors of target nodes and involve higher-order neighbors in the profiling process [40, 41, 42]. Although such models do not fully leverage global structures in the profiling process, they attempt to exploit a larger part of a given network in the process. However, cluster structures as an important global description of network structures is largely ignored in these works. It has been shown that cluster structure is one of the most prominent features of networks, which reveals the generative process of networks and large-scale functional components [19, 43, 44].

Considering the importance of community structures, there is another class of works that analyze the network more comprehensively. These models typically rely on clustering or graph coarsening models to cluster a given network to comprehensively capture global structures [13, 45, 46]. Having the obtained clusters, they mix such structures with the information obtained from local structures to build user profiles. In the following, we describe some representative examples for each class of models and then investigate their effectiveness with respect to *efficiency* and *complexity* challenges on interest-based and interaction-based networks.

2.1 MODELS BASED ON LONGER CYCLE PATHS

Various methods have been introduced in the network mining context that aim to capture global structures in their modeling process using longer cycle paths among users/nodes. A classic example of such methods is link prediction models [42, 47]. Among the different classes of methods for link prediction, the most popular ones are those based on structural-based similarity measures [42]. One of the basic yet popular structural-based similarity measures used for link prediction is the number of common neighbors between target nodes. The extended versions of this measure have also been introduced which penalize high degree common neighbors [48, 49, 50]. The measures based on common neighbors are called local similarity indices as they are solely based on the information of immediate neighbors [42].

However, some works have attempted to consider the overall structure of the network to build similarity measures which are called global structure-based similarity or path-dependent measures/indices [42]. These measures mainly rely on path information between target nodes where paths have lengths of more than two. Kats index is an example of such measures that take into account all of the paths between the target nodes [51]. The other variants of global measures include Simrank [52], matrix-forest index [53] and hitting time [54].

Representation learning models can also be considered as successful methods for user modeling, which have attempted to leverage longer cycle paths [22, 55]. In node embedding, the goal is to encode nodes based on a low-dimensional space where the obtained vectors can capture their graph position. Various techniques have been introduced in the literature for different settings. It has been shown that graph embedding models can be described based upon the framework of encoding and decoding graphs [22]. The encoder-decoder framework suggests that representation learning relies on two main operations. (1) The model uses an encoder to generate the embedding of a node. (2) Then a decoder operator is employed to reconstruct an original graph structure. In most of the graph embedding works named as shallow embedding, the encoder component is a lookup table that maps node ids to their vector representations. However, in a more general setting which is known as Graph neural networks (GNN), the encoder may take node features and graph structure as the encoder's input (rather than the node ids). Having the output of the encoder, the decoder reconstructs certain graph statistics from the node embeddings. Depending on the target graph statistic, different decoders can be developed. However, the classic practice is to use pairwise decoders, which predicts the similarity between target pairs of nodes. The goal of the representation learning algorithm is to mutually optimize the encoder and decoder to minimize the reconstruction loss of the target similarity function.

In this encoder-decoder framework, the similarity function behind a decoder plays a key role in capturing global and local structures of a network [22, 55]. For example, a simple pairwise decoder like the one used in the graph factorization model considers only single length paths, i.e., two nodes are similar if they are connected. This simple decoder solely captures the local structures. However, a decoder may rely on more complex node-node similarity functions that capture both local and global structures. For example, GraRep [56] and HOPE [57] models define their similarity functions based upon the powers of the target adjacency matrix (representing the input graphs) and neighborhood overlap measures, respectively. These similarity functions allow the model to leverage structure beyond immediate neighbors of target nodes in the embedding process, i.e., it determines the similarities of two nodes based on the paths of different lengths between them.

While such methods use deterministic models to capture node's similarities, more recently

stochastic measures of nodes have seen a surge of success [22]. The key innovation behind these models is that two nodes are similar if they tend to co-occur in random walks over a given graph. In fact, these models again rely on longer cycle paths between nodes to capture their similarities; however, they do so in a non-deterministic manner. Popular models like DeepWalk and node2vec are representative of such models [12, 58]. Several methods have been proposed which extend this general idea, e.g., different ideas have been introduced for sampling neighboring nodes to generate random walks [22].

While shallow embeddings have achieved many successes in the past few years by relying on their power in integrating local and global structures, they suffer from some important drawbacks [22]. First, they directly obtain a unique embedding for each node and do not use parameter sharing, which is statistically and computationally inefficient. Moreover, involving node features in the profiling process based upon shallow embedding techniques is challenging. Lastly, such techniques are inherently transductive. That is, they cannot generate embeddings for nodes that do not exist in the training process.

Graph neural networks (GNNs) respond to these limitations by employing more sophisticated encoders [59, 60]. The GNNs can be described based on a neural message passing framework where embedding vectors are exchanged between neighbors and processed/updated using neural networks. In fact, message passing is performed in multiple iterations where in each iteration, an embedding of a node v is updated based on the hidden embedding of its neighbors. That is, the model aggregates messages from v's local neighbors using a set aggregation function. This general framework allows to implicitly involve information about k hop neighbors of a node in its embeddings after k iterations.

Recently some GNN-based models have been proposed to leverage higher-order neighbors in a more explicit and systematic manner [36, 60]. These models propose that in each iteration of message passing, rather than updating the embedding of a node based on its immediate neighbors, messages can be passed from higher-order neighbors of the node. To this end, the k order neighbors were obtained by multiplying the adjacency matrix of connections k time by itself. Also, it has been suggested that the aggregation model can learn how much to involve each layer of the neighbors. It has been shown that this new variant of GCNs has more representation capability than vanilla GCNs since they can represent two-hop delta operators [36]. We can also view hypergraph neural networks as GCNs that try to capture global structures in a more targeted manner [61, 62]. These models rely on clique expansion and the spectral theory of hypergraphs to determine how higher-order message passing should be performed at every message passing step. In fact, hypergraph neural networks have become popular and successful for learning on hypergraph-structured data.

Despite the enormous success of the introduced models, it should be considered they do not fully leverage global structures in their modeling process. In fact, these models only go a few steps beyond the immediate neighbors of a node. Even some of the introduced models like GCN models have limited power in capturing the structures beyond the immediate neighbors of a node. For example, recent successful models like GCN [14] and GAT [40] achieve their best performance when the number of layers is set to two because stacking more layers degrade their performance. A prominent problem with involving higher-order neighbors in GNNs is the issue of over smoothing. That is, as we increase the number of layers, the obtained representations of nodes tend to converge to a certain value and hence become less indistinguishable. It has been shown that residual connections can alleviate a similar problem in computer vision ResNet [63]. However, adding residual connections to the GNNs is not quite effective [64]. More recently, a model named as Dropedge has been introduced that suggests that to relieve the impact of over smoothing, a few edges from the input graph can be removed [65]. While the experiments reveal some improvements over classic GCNs, the obtained improvements are marginal.

2.2 CLUSTERING-BASED MODELS

The idea of involving longer cycle paths in the profiling process is an attempt to partially exploit global/macroscopic structures in the profiling process. However, there is another class of models that incorporate macroscopic structures in a more comprehensive manner by relying on graph clustering/coarsening techniques. This idea has been incorporated in different families of node/user modeling techniques such as feature-based link prediction models [46], shallow node embedding models [13], and graph convolutional networks [45]. It should be noted the notion of clusters can be defined with respect to a target network and a given task of profiling. For example, in a network with a single type of relation and a single type of node, a community is defined as a sub-graph of densely connected nodes with weak ties towards other communities. While in a signed network, a community is defined as a sub-graph with positive intra-cluster connections and negative inter-cluster connections [66, 67]. Hence, the definition of cluster structures depends on a target context/problem. In the following, some representative examples of models that leverage cluster structures are introduced.

As mentioned, an important class of link prediction models rely on features extracted from the common neighbors of target nodes, i.e., similarity measures are defined between target nodes based on their common neighbors [47]. If we rely on individual common neighbors, the obtained similarities can only capture local structures. However, network clustering allows

injecting global structures into the similarity measures. Classically, a cluster in a graph can be defined as a densely connected group of vertices sparsely connected to other clusters [26, 68]. Based on this concept, it has been suggested that the common neighbors of target nodes may carry different meanings depending on if they belong to the same cluster [46]. As such, they define link prediction features for target nodes based on the number of inter-cluster and intra-cluster common neighbors. They show that regardless of the clustering algorithm used to cluster input networks, the link prediction accuracy based on cluster-level features is superior to that based on node-level features [46].

Graph coarsening/clustering has also been used to build more effective shallow representation learning models [13, 69, 70, 71]. For example, in [71], the authors suggest that it is a key requirement for an effective representation learning model to be able to reflect the cluster structures. That is, the embedding vectors of nodes within a cluster should be more similar than those assigned to different clusters. This idea can also aid in alleviating the data sparsity problem. Let's consider two nodes that belong to the same cluster but have weak ties due to the sparsity issue. If we rely only on local structures to embed them, their embedding vectors are expected to have low similarities. However, by injection information obtained from cluster structures, more discriminative/accurate representations can be learned. To implement the proposed idea, a novel Modularized Nonnegative Matrix Factorization (M-NMF) model is introduced, which can consider both pairwise node similarities (similarity-based on local structures) and communities (global structures) [71]. To capture local structures, using the general idea of matrix factorization, first and second-order proximities are preserved (paths with lengths one and two). To involve global structures, communities are discovered based on a modularity constraint term added to classic matrix factorization. Finally, these two terms are jointly optimized.

HARP is another prominent/successful example of leveraging cluster structures in shallow embedding techniques [13]. The model suggests involving cluster structures to set more cluster-aware initializations of node embeddings in the embedding process. In classic shallow node embedding techniques, the embedding vectors are initialized with zeros or random numbers. The authors suggest that by initializing the vectors to cluster-aware initialization, not only cluster-structures can be involved in the embedding process, but also the risk of converging to a poor local mininma is reduced. The proposed model can be viewed as a multilevel paradigm for node embedding. The model considers the hierarchy of network clusters as approximations of an input network at different levels. To obtain a graph hierarchy, a graph coarsening procedure is performed in multiple iterations. Generally speaking, solving the approximation scheme is simpler, i.e., embedding the clusters within a network is simpler than embedding the original network. Hence, they suggest that the representations learned

at a higher level can be used as initialization for the next level. The model can be applied as an add-on to any state-of-the-art shallow graph embedding techniques. The paper confirms the effectiveness of the idea of involving graph structures in the embedding process.

Recently, GCN-based models have also been extended to leverage community structures [45, 72]. For example, in [45], it has been shown that involving cluster structures improves not only the effectiveness but also the efficiency of GCN-based models. GCNs have successful applications in various domains; however, training GCNs on large-scale graphs is challenging due to high memory and computational cost requirements. To alleviate the problem, at each iteration/step of the GCN, the model introduced in [45] samples a set of nodes from a densely connected sub-graph and uses it as a pool to select neighbors from. By applying this simple idea, the model achieves much lower computational and memory costs. The reason for this is that the model uses some *informative* representative neighbors of nodes to build their representations rather than collecting information from the entire set of neighbors, among which some are less relevant neighbors. Moreover, the paper shows that we can achieve higher accuracy in tasks like node classification by training deeper GCNs than baselines. Indeed, training deep GCNs using vanilla GCN is practically infeasible due to high computation.

2.3 LIMITATIONS OF THE EXISTING APPROACHES

As discussed, two important classes of approaches have been introduced to bridge local and global structures for user modeling purposes. While methods based on higher-order neighbors do not fully leverage global structures, models that rely on clustering algorithms leverage global structures in a more comprehensive manner. However, it should be considered in both classes of approaches the computational cost of user modeling for certain applications is infeasible. Both classes of models require incorporating the entire set of edges within a network to build their models. However, for some applications of user modeling that require an immediate response from the modeling method, analyzing an entire network has burdensome computations. As such, it can be said that for some scenarios of user modeling, the existing local-global structure-based modeling methods encounter the *efficiency* challenge.

Moreover, it should be considered that the proposed models are mainly focused on classic networks (with single edge type). However, users may develop multi-type of relationships in social networks and online systems. Recently, some efforts have been devoted to addressing the *complexity* challenge caused by multi-type of interactions. However, their effectiveness in approaching the challenge is limited. For example, some works have attempted to extend shallow embedding techniques and GCN-based models to perform on networks with multi-

types of interactions [22, 73, 74, 75]. However, the models have higher complexity than vanilla models which makes training them harder. Moreover, even the proposed extensions do not effectively leverage global structures, i.e., they make some strong assumptions to take into account global structures.

CHAPTER 3: GRAPH-BASED HASHTAG RECOMMENDATION

3.1 OVERVIEW

Interest profiling of users and personalization of services play a crucial role in the usefulness of OSNs. In particular, with an overwhelming amount of information spreading in our digital universe through websites like Twitter and Weibo, having an effective personalization system to deliver information aligned with user's interests is of great importance. In this chapter, we investigate the problem of hashtag recommendation on microblogging websites, which can be regarded as a representative of the interest profiling task for personalization purposes. In fact, the model introduced in this chapter can be adapted to address other recommendation/personalization tasks on microblogging websites like user-to-user recommendation.

Tagging is an important technique for organizing and retrieving items in information systems, i.e., tags make tagged items more findable by users and also aids to organize collections of items. A successful instance of tagging technique is the notion of *hashtag*, which has proven remarkably effective for labeling microblogs. While the usefulness of hashtags has been exploited for many applications (e.g., [76]), a huge and dynamic pool of hashtags exists in microblogging websites, which could undermine the effectiveness of the notion of hashtags. As a result, it has been shown that only a small portion of microblogs are annotated with hashtags [77]. Therefore, the task of Hashtag Recommendation for Microblogs has become indispensable for microblogging websites.

The problem can also be investigated from another perspective: for a user who wants to follow hashtags to find microblogs aligned with her interests, discovering relevant trending hashtags could become a challenge. Although successfully addressing the problem could substantially promote user satisfaction and engagement, it has not received much attention so far in academia. A possible approach could be building content-based models, i.e., given the microblogs of a user, the content of the microblogs can be analyzed to identify her interests [78]. Or we could employ classic Collaborative Filtering (CF) models to make recommendations by analyzing historical user-hashtags interactions [79]. However, only a small portion of users tend to generate abundant content data and interact with hashtags which strongly affects the applicability of such approaches [7].

Problem: Given a user on a microblogging website like Twitter, how can we recommend her hashtags, in particular when sufficient content/interaction data is not present?

We propose that the problem can be approached by relying on graph data. In fact, it has

been shown that 1) a portion of links can be regarded as interest-based links [5, 27], 2) and also link data is more abundant than content data on such websites since most of the user tend to be 'listeners' [7].

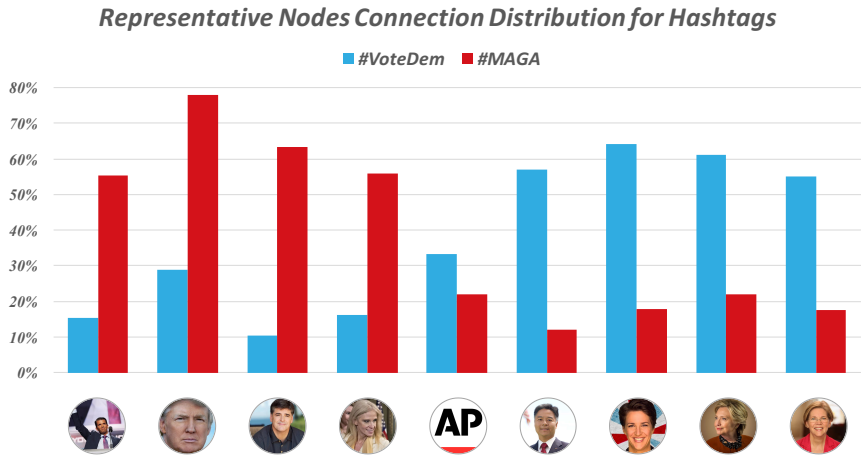

Figure 3.1: The percentage of links toward 10 representative nodes in the US politics for two groups: users who tweeted hashtag #VoteDem and users who tweeted #MAGA.

However, *how do we exploit users' social linkage in an effective way to address the problem?* A classic approach could be profiling a set of users in the network based on their content/interaction data and then propagating their embeddings in the network to their immediate and higher order neighbors through recent methods like graph convolutional networks [15] or graph attention networks [80] to build the profiles of other users. In fact, this method requires us to go beyond local neighbors of the seed set and visit the entire network. Nevertheless, such methods do not fit our recommendation problem due to two major reasons:

Efficiency: It is computationally expensive to apply them in multiple iterations to a huge network like Twitter. In fact, even crawling a network with millions of nodes and billions of edges has burdensome computations. Also, microblogging websites are highly dynamic and a large number of nodes/edges emerge on a daily basis which means that the profiles of users need to be updated frequently. Hence, a profiling method that requires to crawl the entire network may need burdensome computations.

Effectiveness: Such models build a fixed profile for each user. In a microblogging website, a user may have diverse and independent interests such as politics and sports. To determine whether a hashtag related to politics is relevant to the user, it is more effective to build a profile that focuses on her political interests, i.e. , involving unrelated interests can potentially add noise in the prediction process. In fact, this is the core idea behind most of the

CF-based models, including item-based CF [81].

Towards building a model that addresses the mentioned drawbacks of a classic graph-based encoding technique, we aim to take advantage of the structural characteristics of the microblogging networks. In fact, our idea is that we can use the local structures around labeled users as bridges to reach to the rest of users in the network, i.e., the goal is to bridge the local and global structures. Microblogging networks are scale-free graphs [7, 17, 18]. There are a set of hub nodes that receive a large portion of the other users links [17]. Also, it has been shown that hubness/popularity of a user on Twitter indicates that she represents a topic of interest [7]. That is, the links towards hub nodes effectively capture user's interests. As such, our insight to build an efficient model is to focus on user's links towards hub nodes rather than analyzing the entire network. Fig. 3.1 illustrates the intuition behind the informativeness of hub nodes in determining what hashtags a user might be interested in. It depicts the distributions of the links that two sets of users have towards a small set of hub nodes in the US politics in which followers of a republican leader are more likely to use hashtag #MAGA while the followers of a democrat leader tend to use hashtag #VoteDem.

Based on this insight, our key idea is that in a microblogging website, the embeddings of nodes can be derived by aggregating the embeddings of the hub nodes they are connected to. That is, rather than finding the embeddings of entire nodes in the network, we can obtain the embeddings of hub nodes and build a function to transitively embed other nodes. We use this general idea as the basis of our hashtag recommendation model, i.e., we pick a set of hub users denoted as **representative nodes** and project those nodes and hashtags into a shared latent space. Given a user and a hashtag, we construct an embedding of the user based on her representative followees and then feed it to a scoring model along with the embedding of the target hashtag to generate the relevance score.

The proposed architecture meets the efficiency challenge. It relies on profiling a very small portion of the nodes and transitively derives the profiles of other users. Also, it deals with the high dynamics of the network. Unlike ordinary users, profiles of representative-nodes have low dynamics, and the dynamics of the emergence of such nodes are quite slow. The embeddings of users are defined as *functions* of representative-nodes profiles. Hence, we can capture the frequent changes in user's profiles by simply changing the function's inputs. Moreover, the model is inductive as it can make recommendations for unseen/new users.

More importantly, it enables us to use hashtag-aware embeddings of users in the recommendation process, hence addresses the effectiveness challenge. To embed a user based on her representative followees, we propose a model based on attention mechanism which has been successfully employed in different problems [81, 82]. Given a hashtag, our proposed attention model embeds the target user by performing a weighted aggregation of her followees'

embeddings where the weights reflect the relatedness of a hashtag to a representative node.

However, employing the idea of attention mechanism in the proposed model is challenging. Classically, an attention model can be learned implicitly, i.e., in an end to end manner based on the final objective of the model. However, due to sparsity issue caused by a large number of free parameters in our model (a huge pool of representative nodes and hashtags), the attention model learned by implicit training generates low quality attention maps. To tackle this issue, as a general approach, explicit supervision can be injected into attention models using labels specifically acquired for the attention model [82]. However, the challenge we encounter is that the labels for relevance of representative nodes to hashtags do not exist. In response, we propose the idea of weak supervision, i.e., using statistical methods, we first generate weak labels for the relevance of hashtags to representative nodes and then use the labels as explicit supervision for the attention component.

Lastly, to show the flexibility of the proposed graph-based model, we further develop the model to accommodate content (text) data. In fact, by leveraging content data along with graph data we can make more accurate recommendations for those users associated with content data. We conducted comprehensive experiments on two datasets obtained from Twitter and Weibo and compared the performance of the model with multiple baseline recommendation models. Our results show that the proposed model substantially outperforms the baselines in terms of NDCG and HR. Also, the results confirm the considerable effectiveness of the proposed supervised attention-based technique over the unsupervised version of the model. The major contributions of the work can be summarized in three perspectives:

- We introduce the problem of trending hashtag recommendation based on graph data.
- We introduce the novel idea of node embedding based on representative/hub nodes which can exploit both local and global structures and is effective for large scale dynamic graphs like Twitter or Weibo.
- We propose the notion of weakly supervised attention in a graph setting to effectively encode users based on their neighbors in the context of a hashtag or topic.

3.2 RELATED WORKS AND PRELIMINARIES

3.2.1 Related Works

We review the studies that are related to our work. In fact, the related works of our work can be described in three different aspects. We first describe recent literature on the problem

of Hashtag Recommendation. Next, we describe how our work is related to the topic of social recommendation. Finally, we review the literature related to Attention Mechanism.

Hashtag recommendation. Although hashtag-recommendation for *users* is an important task for user engagement in Microblogging website, it has not received much attention in the literature. To the best of our knowledge, [79] is the only work that attempts to address the problem. The model feeds the user-hashtag interaction matrix to a classic Matrix Factorization model to make recommendations. Clearly, the effectiveness of the model is quite limited, especially for users with zero or sparse interactions.

While our focus is on graph data (UHR task), the proposed model (when graph and text data are combined) can serve as a personalized hashtag recommendation model for Microblogs. The existing model for MHR can be categorized into two groups: 1) Traditional global models, 2) Personalized models

1) Vast majority of existing models for *hashtag recommendation for microblogs* address the task by only analyzing the content of the target Microblog. For example, classic methods such as tf-idf based retrieval models and translation-based methods have been adopted to address the task [83, 84, 85]. Deep Learning-based hashtag recommendation models have also recently emerged. [86, 87, 88, 89].

2) Some works also have tried to improve classic text-based models by making personalized recommendations. Zhang *et al.* extended upon the translation-based methods to leverage personal factors [90]. Analogously, in [91], user related features including tweeting-histories, location, and social influence were incorporated as auxiliary information to develop a personalized model. Also, recently personalized hashtag recommendation models for multimedia microblogs has been introduced [92]. Rawat et al. considered the contextual preferences of users on images (e.g. time and geo-location) to build user representation and use it in hashtag recommendation for images [93]. Park et al. build a model of user preferences based on their most frequently used hashtags and devised a Context Sequence Memory Network for hashtag recommendation for images based on user preferences and image features. [94].

While these works mainly focus on users with dense interaction history, our model can cover all of the users including those with sparse interactions by modeling them solely based on graph data.

Social recommendation. The problem of hashtag recommendation based on user's connections on microblogging networks can be viewed as a social recommendation problem. The social recommendation problem is defined as any recommendation that incorporates the social relations of users as an additional signal to make recommendations [95]. Various social recommendation models have been introduced in the literature that adopt different classes of recommender systems. They mainly rely on different variants of collaborative filtering

algorithms as their basic models. In fact, it has been shown that social recommendation models can be categorized based on the type of collaborative filtering method they use as their basic model [95]. For example, social-based weight mean models use memory-based CF as their basic model [96]. These models first collect a set of *correlated* users for the target user and then aggregate their ratings towards the target item to predict the missing rating. The models differ based on how they determine correlated users based on social relations [97, 98].

Another class of models is those based on model-based CF methods [99, 100, 101]. These methods first determine the strength of the relations between users based on the similarity of target user's ratings. Next, they involve the obtained relations for matrix factorization by defining a new term in the matrix factorization framework, which captures the social relations. Co-factorization methods [101, 102], ensemble methods [100], and regularization methods [99] are social recommendation models that rely on the matrix factorization framework.

While the introduced social recommender systems, including memory-based and model-based methods, use different approaches to involve social relations, they share a key principle: a user and her neighbors are expected to have similar ratings towards items. In fact, this key assumption relies on the notion of homophily in social relations. However, our strategy differs from the existing techniques in different aspects. In our model, we only rely on hub neighbors of users rather than the entire set of their neighbors. We use hub nodes as entities that directly reflect user's interests rather than using them as proxies to make predictions. Moreover, we assume that not all of the hub neighbors of a user need to be *similar* to the user with respect to all of her interests. That is, a hub node may represent a user only from a certain perspective (interest topic). Lastly, most of the social recommendation models rely on the target users' previous interactions with the item set to determine how her social neighbors should be involved in determining the taste/preference of the user. However, in our setting, a user may have no interactions with item sets and her preference can be solely determined based on her social connections.

Attention mechanism. In general, to compute an upper-layer of the neural network, the attention mechanism can be used to determine a weight to each position in a lower-layer [103]. Attention-based models have been successfully applied for a wide range of sequence-based tasks including text entailment [104], image captioning [105], sentence summarization [106] and recommender system [81, 82]. For example, a recent successful work in graph mining context has introduced an attention-based architecture to perform node classification of graph-structured data where the hidden representations of each node in the graph is computed by attending over its neighbors [80]. Also, a recent work in recommender systems

has introduced an attention network which is capable of distinguishing which historical items in a user profile are more important for a prediction [81].

Classically, for models with low complexity and in presence of abundant training data, the attention components can be trained in an implicit manner based on the final output of the model [80, 81, 107]. However, that is not the case for model with higher complexity. In fact, some recent works in computer vision and language translation have shown that the accuracy of the attention map generated by an attention model learned in an implicit manner is not ensured [94, 108, 109]. For example, [109] shows there is a substantial gap in quality between the alignments obtained from the classic attention models and the human labeled alignments. To tackle this issue, in computer vision, explicit supervision has been injected to attention models using labeled data specifically acquired for the attention model [82]. We follow the same core idea in our attention model, however, we devise the notion of weakly supervision based on statistical models.

3.2.2 Preliminaries

We first formalize the problem and then shortly recapitulate the widely used Generalized Matrix Factroization model which is the basis component of our proposed model.

Problem formulation. Let $G(V, E)$ denote the user's directed connections in a Microblogging website where V is the set of nodes/users, and E represents the set of the links. And let U denote the target set of users with the size M where $U \subseteq V$, H the set of hashtags with size N, and T_u the set of the tweets of user u. We define the user-hashtag interaction matrix as $\mathbf{Y} \in R^{M \times N}$ where $y^{uh} = 1$ if an interaction is observed between u and h, e.g., u has used h in one of his tweets T_u, and $y^{uh} = 0$ otherwise. Note that a value 0 does not necessarily mean u is not related to h, it can be that the user is not aware of the hashtag or the user is not active generally. Also, it should be mentioned, unlike a traditional recommendation problem, a large proportion of users within U do not have any interactions with H. The recommendation problem for $u \in U$ is formulated as the problem of estimating the scores of unobserved entries in the corresponding row of $\mathbf{Y}$, which are used for ranking the hashtags for u.

Generalized matrix factorization. Matrix Factorization can be considered as the most popular model for recommendation and has been investigated extensively in literature [110]. Matrix Factorization based models define the relevance score of user u and item i as the inner product of their latent vectors $\hat{y}_{ui} = p_u^T q_i$ where p_u and q_i represents the latent vectors of u and i respectively. Generalized Matrix Factorization is a recent model that uses a multi-layer neural-network structure to mimic classic MF [111]. In GMF, the input layer consists of two

feature vectors v_u and v_i that describe user u and item i, respectively. Above the input layer is the embedding layer; it is a fully connected layer that projects the sparse representation of the target user and item to dense vectors. The obtained user/item embedding can be seen as the latent vector for user/item. The user embedding and item embedding are then fed into a neural architecture to estimate the relevance scores.

3.3 PROPOSED MODEL: PHAN

In this section, we describe the structure of the proposed model denoted as Personalized HAshtag recommendation based on Netwrok data (PHAN). In the following, first, the notion of representative nodes is illustrated as the core idea of the model; next, the model is introduced in a top-down manner: we start with the prediction model, and then the encoding techniques are described; finally, we explain how the model can be altered to capture both text and graph data. Fig. 3.2 represents the structure of the model based on both graph and text data.

3.3.1 Representative Nodes As User Features

To predict the relevance of user u to hashtag h, we rely on GMF [111]. The feature vector of users and items in GMF can be customized to support a wide range of modeling of users and items, such as content-data, user-item interaction, or context data. In our problem setting, content-based or interaction-based features are not applicable because most of the users are 'listeners'.

We propose to exploit user's links on graph G to profile users based on the fact user's links implicitly indicate their interests, and also graph data is more abundant than content data [5]. However, it is of great importance to build the model in a way that it is *efficient* and *effective*. As mentioned, a general graph embedding technique does not meet this criterion. Hence, we aim to address the problem by introducing the notion of *representative nodes*.

Representative nodes as cluster centers of interest-based clusters. There are two main reasons for the creation of a link from an initiator node to a receiver node in a microblogging website. 1) The receiver node is associated with a topic interesting for the initiator node. 2) There exist a friendship/social tie between the nodes [5, 7]. A user can receive only a limited number of social tie-based connections. This implies that when a node becomes popular in a microblogging network, it boldly represents a topic of interest. In fact, there exist some popular representative node for each topic of interest, and if a user is

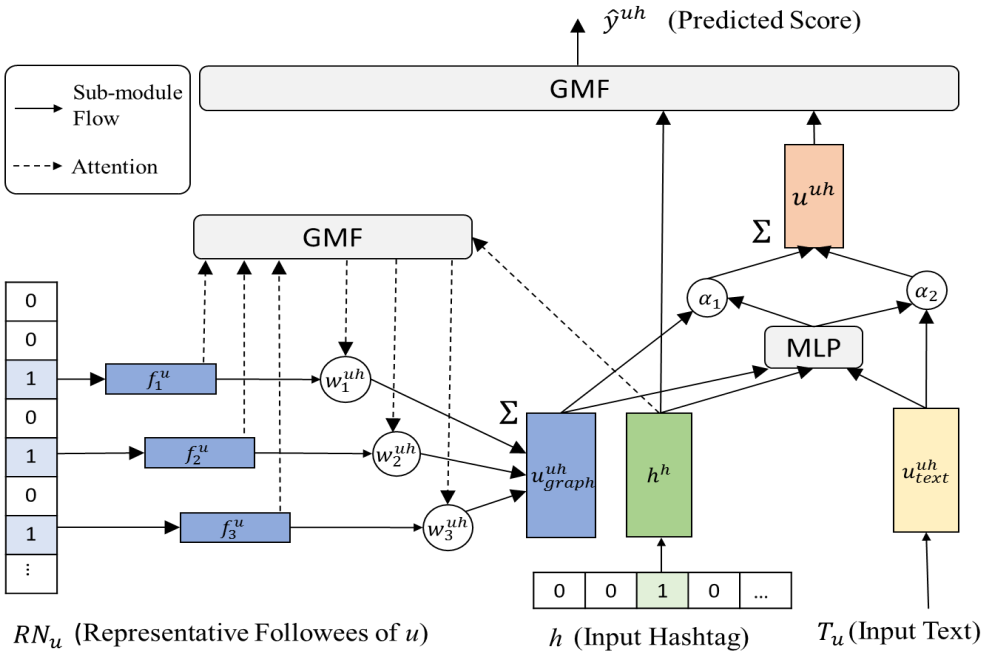

Figure 3.2: Architecture of PHAN based on graph and text data. The embeddings of the target hashtag, h^h, and the embeddings of the u's representative follows, RN_u, are obtained from lookup tables. The graph-based embedding of u with respect to h, u^{uh}_{graph}, is the weighted aggregation of embeddings of u's representative followees where the weight of the embedding of node j, w^{uh}_j, is determined by feeding f^u_j and h_h to the GMF function. u's text-based profile, u^{uh}_{text}, is built by an off the shelf embedding model. u^{uh}_{text} and u^{uh}_{graph} are combined in a weighted way to build a unified profile of the user, u^{uh}, where the weights are determined using an MLP. Finally, $\hat{y}^{uh}$ is generated by feeding h^h and u^{uh} to the GMF.

interested in a topic, she very likely follows some of the corresponding representative nodes [18].

Based on this generative process of links in microblogging websites, we observe that Microblogging networks are scale-free networks, i.e, they contain some hub nodes and a large portion of links are toward hub nodes [18]. Also, those links are noiseless and purely indicate user's interests (unlike the links among ordinary users which are likely to represent friendship ties) [5]. In accordance with the power-law distribution, the number of hub nodes is extremely smaller than the number of nodes in the network [17].

In other words, it can be said that for each cluster of users interested in a topic, there exist a set hub nodes that can be regarded as the centers of the cluster. Note that unlike the classic definition of a cluster where clusters are defined as a set of densely connected set of nodes, in **interest-based clusters**, nodes within a cluster tend to have connections towards the corresponding cluster centers while are not necessarily densely connected to each other.

Also, cluster centers have slow dynamics. They normally appear or change in the network gradually over time. Moreover, the interest topics associated with a representative node are mostly fixed except for rare cases, e.g., when a popular athlete becomes a politician. Lastly, it is practically straightforward to identify topics represented by cluster centers and profile them as they are associated with ample data (both content and graph data).

These characteristics of representative nodes make them a noiseless, low dynamic, and dense data source in general to profile users and in particular, to address our target problem. In fact, first we can profile hub/representative nodes within a network and then we can profile users based on the profiles of the cluster centers they are connected to. This idea allows us to leverage global structures of target graphs because hub nodes indeed represent global structures. It can be said that based on the proposed idea, we identify the structure of a given network, i.e., determine the clusters of the network and profile each cluster within the network. We emphasize the idea of user profiling based on representative nodes heavily depends on the fact that the input network is a scale-free network.

Selecting representative nodes As the initial step of the model, the set of representative nodes is selected. Given the target set of users U and a graph $G = (V, E)$, we define representative nodes $RN = \{r_1, r_2, ..., r_n\}$ as the set of nodes in V with at least L followers in U where n is the number of representative nodes. Accordingly, the set of representative followees of the user u is represented as $RN_u \in RN$. It should be noted, L is a parameter of the proposed model to be tuned according to desired performance and efficiency. In general, the value of L is decided by the size of U. We further discuss this in the experiments section.

3.3.2 Predicting Relevance Score

Following the structure of GMF, given the embedding vector of the target user $\mathbf{u}^{uh}$ and the target hashtag $\mathbf{h}^h$, the relevance between u and h is computed as follows:

$$\hat{y}^{uh} = \sigma(\mathbf{W^T}(\mathbf{h^h} \odot \mathbf{u^{uh}}) + b), \tag{3.1}$$

where σ is Sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$, $\mathbf{W}$ and b_{fc} are the weight and bias of GMF, $\odot$ denotes element-wise multiplication. Note that, the GMF function ca n be replaced with other prediction models such as multi-layer perceptron (MLP) as in [111]. In the next subsections, the embedding components of the model are introduced.

Hashtag embedding: In our problem settings, hashtags are not associated with any extra data but identities. Hence, we represent hashtags as one hot vectors of identities and learn their representations by building a lookup table which can be learned in our training

process. Note that additional data can also be incorporated to embed hashtags.

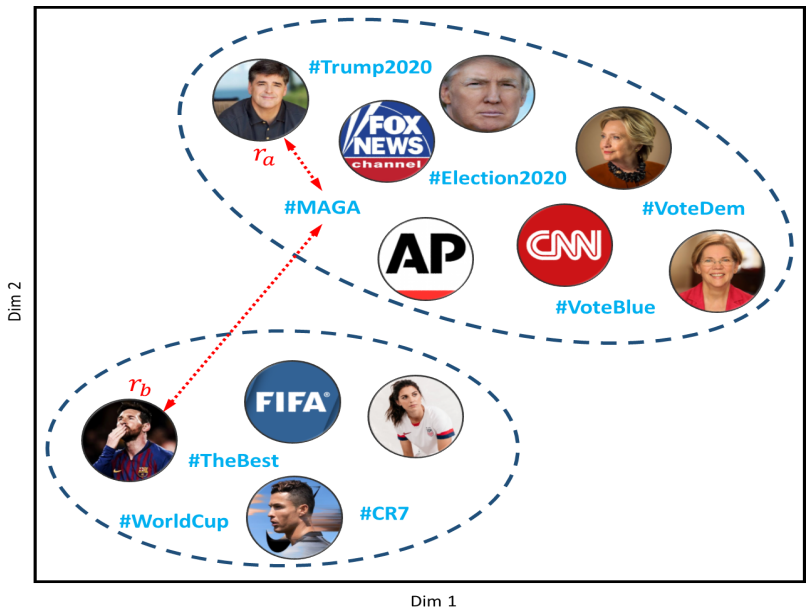

Figure 3.3: A toy example representing the hashtags and representative nodes related to politics and soccer in a shared latent space.

3.3.3 User Embedding based on Weakly Supervised Attention

We propose a weakly supervised attention-based graph embedding technique to embed user u based on RN_u to determine its relevance to hashtag h. In the following, we first introduce a basic approach to encode the followee list of users to better motivate the proposed model and then illustrate the architecture of the model.

Design 1: fixed embedding. A basic approach to encode a user based on her following list is to create a one-hot encoded vector where each element corresponds to a representative node and is marked 1 if the user follows the node and 0 otherwise. The resulting sparse vector can be projected to a fully connected dense embedding vector.However, this trivial approach is oblivious to the target hashtag information and creates a single fixed representation for the user. A user might have a set of independent interests, e.g., sports, politics and music. When predicting the relevance of a user to a hashtag, it is more accurate to focus on the part of the user's interests that is related to the target hashtag.

Design 2: embedding based on classic attention. Towards building a hashtag dependent representation of a user, we suggest that hashtags and representative nodes can be projected into a shared latent space. Roughly speaking, we assume if user $u \in U$ follows

representative node r and uses hashtag f in one of his tweets, it indicates proximity between r and f in the embedding space. Having the embeddings of representative nodes and hashtags, the hashtag dependent embedding of the target user can be obtained by aggregating the embeddings of her representative followees in a weighted way where the weights are determined based on the relevance of the representative nodes to the target hashtag.

We further explain the idea behind our target aware embedding model using the toy example depicted in Fig. 3.3. The figure shows the representative nodes and hashtags of two topics: politics and soccer in a shared latent space. Let's assume the user u follows some representative nodes related to politics, including the representative node r_a and some representative nodes related to soccer, including the representative node r_b. Suppose the task is to determine the relevance of u to hashtag #MAGA. We suggest that to embed u with respect to #MAGA, the attention model should attend more to the representative node r_a than r_b because r_a has higher proximity to hashtag #MAGA in the embedding space. In this way, the model focuses on the politic related profile of the user to make the prediction and filters out the unrelated parts.

More formally, the attention-based model can be described as follows: let's assume we have a lookup table for the embeddings of representative nodes which can be learned in an end-to-end manner based on the final output of the model. We can look up the embedding table and get the embeddings of the representative nodes followed by the user u denoted by $\mathbf{F}^u \in R^{s\times n}$ and hashtag embedding denoted by $\mathbf{h}^h \in R^{s\times 1}$, where s is the size of the embedding vector, and n is the number of representative nodes followed by u. We define the user embedding as:

$$\mathbf{u}^{uh}_{graph} = n^{-\beta} \sum_{j \leq n} \hat{w}^{uh}_j \cdot \mathbf{f^u_j}, \tag{3.2}$$

where β is a normalization parameter ranging from 0 to 1 and $\mathbf{f^u_j}$ represents the embedding of the j-th representative node in $\mathbf{F}^u$, i.e., $\mathbf{F}^u =< \mathbf{f^u_1}, ..., \mathbf{f^u_n} >$. Also $\hat{\mathbf{W}}^{uh} =< \hat{w}^{uh}_1, ..., \hat{w}^{uh}_n > \in R^{n\times 1}$ is the set of attention weights for user u where $\hat{w}^{uh}_j$ denotes the attention weight of the j-th representative node with respect to hashtag h.

We employ GMF-based prediction model to compute the attention weight for each representative node given the target hashtag. As such, given a representative node and a hashtag, the dependency between them can be computed as follows:

$$\hat{w}^{uh}_j = \sigma(\mathbf{W^T}(\mathbf{h^h} \odot \mathbf{f^u_j}) + b). \tag{3.3}$$

It should be noted that the GMF model we use in our attention model shares the same parameters as the GMF model for relevance prediction in because the target user and repre-

sentative nodes share the same embedding space, hence parameter sharing can be used for the GMF models.

Having this structure, the attention model, and the embeddings of representative nodes can be trained in an end to end manner based on the final objective.

However, the success of the proposed graph embedding technique and, consequently, the accuracy of the prediction model depends on the effectiveness of the attention model. The more accurately attention determines the relatedness of representative nodes to a given hashtag, the more accurate encoding of the target user can be obtained. However, according to our experiments, the accuracy of the second design is not superior to the first one. We conjecture the reason for this is that the proposed attention model does not accurately determine the relatedness weights.

Indeed, attention-based models have been used for various problems such as image captioning and Visual Question Answering (VQA) [94, 108], e.g., in VQA, attention is used to selectively target related areas of an image to a given question. Classically, the attention mechanism can be trained in an end to end manner based on the final objective of the model. Generally, this form of training is effective for models/problems with low complexity. However, the accuracy of the attention model that is implicitly trained is not ensured, especially for models with high complexity. For example, in an inherently complex problem like VQA [108], attention models trained implicitly do not focus on the same regions as humans do, which results in incorrect answers.

Why classic attention does not generate accurate attention maps? The proposed attention model can be considered as a sub-component of the main prediction model. When training the model in an end-to-end manner, the parameters of the attention model are learned based on the final output. However, this way of training is quite vulnerable to the vanishing gradient problem [112]. In general, the vanishing gradient problem occurs during back-propagation in a deep neural network when the magnitude of gradients diminishes as we go down the layers. This causes convergence to occur extremely slowly as the updates are highly diminished. Also, overfitting is another issue associated with attention models. Due to the large size of the set of hub-nodes and hashtags, we encounter overfitting/sparsity in classic-attention. Some complex attention-models used in VQA had also faced the problem [108]. In our model, if the embeddings of users are obtained by equally attending over representative nodes (fixed embedding, without target aware attention), the model would have less complexity with a lower degree of freedom. However, applying the attention component to the model adds up to its complexity. Hence, training the model would become more prone to overfitting.

Design 3: embedding based on weakly supervised attention. In order to tackle

this challenge, some recent works have tried to add supervision to the attention component of their models. For example, in a recent VQA model, when human-labeled training data is used for supervising the attention component, not only the accuracy of question answering increased, but also the error of the attention model is reduced [108, 109]. Inspired by the success of the general idea of supervised attention, we aim to add attention supervision to our model. However, we face a major challenge: the labels of relevance for a representative node and a hashtag do not exist.

In response, we introduce the idea of *weak supervision for attention mechanism*. We propose *weak labels* for the relevance of nodes and hashtags based on training data and by relying on statistical models. We introduce two types of supervision based on this idea and inject them into the final loss function of the model. This allows us to jointly supervise the attention and relevance estimation models in an explicit way.

Co-occurrence-based supervision. The main idea behind our first weak supervision is that if a user uses hashtag h in her microblog and also follows r_i, it reflects a relevance between h and r_i. That is, we regard the attention component as a link prediction model: if a user is associated with hashtag h, how likely she follows the target representative node? Accordingly, the desired output of the prediction model w_j^{uh} in $\mathbf{W}^{uh} =< \hat{w}_1^{uh}, ..., \hat{w}_n^{uh} > \in R^{n\times 1}$ is defined as $P(h|j)$, whose Maximum Likelihood Estimate is as follows:

$$w_j^{uh} = P(j|h) = \frac{c(h,j)}{c(h)}, \tag{3.4}$$

where $c(h,j)$ is the co-occurrence count of hashtag h and representative node j in the training data and $c(h)$ is the frequency of hashtag h in the training data.

Informativeness-based supervision. This form of weak supervision is inspired by the concept of TF-IDF in information retrieval. We suggest that the relevance of representative node r_i and hashtag h is decided by two factors: 1) the number of users associated with hashtag h who also follow r_i. 2) the number of users who follow r_i and are associated with hashtags other than h. Through this idea, we penalize those popular nodes that are associated with a large number of hashtags. Following this idea, w_j^{uh} is defined as:

$$w_j^{uh} = \frac{c(h,j)}{c(h)} * \log \frac{c(j)}{c(h',j)}, \tag{3.5}$$

where $c(h',j)$ is the number of times node j has occurred in the dataset but in absence of hashtag h while $c(j)$ denotes the number of times node j occurs.

3.3.4 Loss Function

As mentioned, we inject supervision to the attention model through our loss function. Hence, the loss function of our proposed model consists of two parts: the main loss function and the supervised-attention-based loss function.

Main loss function: The main loss function is defined based on the difference between the estimated relevancy score $\hat{y^{uh}}$ and ground-truth label information y^{uh}. In our task, y^{uh} is 1 or 0 according to whether the user u has interacted with hashtag h. We define the loss function as the Binary Cross-Entropy Loss Function.

Supervised-attention-based loss function: We introduced Attention supervision to guide our GMF model to better estimate the relevance between the target representative node and hashtag, where the supervision information w_j^{uh} in $\mathbf{W}^{uh} =< \hat{w}_1^{uh}, ..., \hat{w}_n^{uh} > \in R^{n\times 1}$ is represented as $P(j|h)$, which means given a hashtag h, how relevant the followee j is. We define the loss function between w_j^{uh} and $\hat{w}_j^{uh}$ as the Binary Cross-Entropy Loss Function as well.

Joint loss function: We define the joint loss function as:

$$\begin{aligned} Loss =& L_{main}(\hat{y}^{uh}, y^{uh}) + \lambda L_{attention}(\hat{\mathbf{W}}^{uh}, \mathbf{W}^{uh}) \\ &= -(\hat{y}^{uh} \log y^{uh} + (1-\hat{y}^{uh}) \log(1-y^{uh})) \\ &- \frac{\lambda}{n} \sum_{j \le n} \hat{w}_j^{uh} \log w_j^{uh} + (1-\hat{w}_j^{uh}) \log(1-w_j^{uh}), \end{aligned} \tag{3.6}$$

where λ is a trade-off setting to show how important the role attention supervision plays in this model. When $\lambda = 0$, the model can be regarded as a classic attention model.

How joint loss function improves the effectiveness of the attention model? 1) By providing explicit supervision for a shallow sub-component of the model, our approach helps to mitigate the vanishing gradient problem. 2) Further, attention supervision acts as a regularizer in the loss function. Hence, it can help prevent overfitting in training the model. Indeed, it adds more constraints to the parameters of the attention model. 3) To build a profile of a user, it is more reliable/robust to attend to representative nodes with more reliable embeddings. The supervised attention-based loss function automatically lowers the contribution of less reliable representative nodes. Statistically, the embedding obtained for a representative node is more reliable if it has been obtained based on more data, i.e., if it has a larger number of co-occurrences with the set of hashtags. According to equations 3.4 and 3.5, in average, less reliable representative nodes will have less relevance to the hashtags. As such, they have a smaller contribution in building the representation of a given user.

3.3.5 Fusing Content and Graph Data

Although most of users are not associated with content data, for active users, abundant content data is available, which can be used to profile them. While we focus on exploiting graph data– as it is orthogonal to text/content data– the knowledge extracted from text data can be fused with the profile obtained from the target user's connections to build a unified recommendation model for users. Furthermore, fusing text and graph data enables us to use the model as a personalized hashtag recommendation model for *microblogs*. That is, the knowledge extracted from a user's microblog can be combined with her graph-based profile to make personalized recommendations for the target microblogs. In fact, having the profile of a user can aid us to better understand the content of her microblogs and resolve their inherent ambiguities, which consequently improves the quality of recommendations.

In both of the scenarios, User-Hashtag recommendation (UHR) and Microblog-Hashtag recommendation (MHR), the recommendation problem can be defined as finding the relevance score between a hashtag h and a pair of text-followee list $< T_u, RN_u >$. Note that if we use the model as a personalized hashtag recommendation model for microblogs, text data would involve a single microblog, i.e., $T_u = \{t\}$ where t is the target microblog.

Text/content and graph data are two different data sources in nature: the former is sequential while the latter is associative (non-sequential). Considering their inherent differences, we propose to encode text and followee data separately and then fuse them to build an embedding of the input data, which can be fed to the prediction model described in Eq. 3.1 to solve the target relevance prediction problem. Various techniques have been introduced in the literature to combine embedding vectors [113]. Based on our experiments, we opt to combine the two output vectors using weighed element-wise addition. We chose to infer the weights using the hashtag information based on the intuition that the importance of the two channels differs depending on the hashtag.

In more details, given hashtag h the unified embedding vector denoted by u^{uh} is obtained as follows:

$$\mathbf{u}^{uh} = \alpha_1 \mathbf{u}^{uh}_{text} + \alpha_2 \mathbf{u}^{uh}_{graph}, \tag{3.7}$$

where α_1 and α_2 are hashtag-dependent weights and $\mathbf{u}^{uh}_{text}, \mathbf{u}^{uh}_{graphl}$ are text and followee list embeddings respectively. To predict the weight scalars α_1, α_2 with regard to the input hashtag, we employ a multi-layer perceptron (MLP) as the weight prediction model with hashtag, text and followee list embeddings as input.

In previous subsections, we described how u^{uh}_{graph} can be obtained. Now the question becomes how to encode the text data? Our fusion architecture allows us to adopt the existing text embedding techniques. We apply two state-of-the-art microblog embedding techniques

to exploit textual information. For the task of Microblog-Hashtag recommendation(MHR) , we apply the classic Long-Short-Term Memory (LSTM) [114] model to encode the single input microblog, which is a widely used technique for embedding short text data. Meanwhile, for User-Hashtag recommendation(UHR), the end-to-end Memory model [115] is used due to its excellent performance in handling multiple microblogs.

3.4 EXPERIMENT

We conducted experiments to verify the effectiveness of our proposed model. We focus on answering three key questions:

- **RQ1:** How does our model perform compared to the baseline methods on UHR and MHR tasks?
- **RQ2:** How effective is the proposed attention model?
- **RQ3:** Do users have sufficient number of links towards representative nodes?

3.4.1 Experimental Setup

Dataset: We experimented with two real-world datasets: Twitter Dataset and Weibo Dataset. Related statistics are described in Tab 3.1 where **RN** denotes Representative Node.

Twitter dataset: Based on a Kaggle dataset[1] that contains Twitter user's followee list, we use Twitter API to crawl users' public tweets from Jan-2016 to Jun-2016 so as to construct the Twitter Dataset. We selected hashtags whose frequency is more than 30 and representative nodes who are followed at least by 120 users.

Weibo dataset: We used the public Weibo dataset[2] collected from Sina-Weibo for evaluation. Similar to the Twitter Dataset, we retained hashtags with frequency more than 6 and representative nodes with at least 40 followers.

Table 3.1: Statistics of the Datasets.

Dataset	**#Tweet**	**#Followees**	**#RN**	**#Hashtag**	**#User**	Avg(#**RN**$_{user}$)
Twitter	217965	253818	9309	2873	23169	33.42
Weibo	10521	585475	8426	2017	7023	121.98

[1]https://www.kaggle.com/hwassner/TwitterFriends/home
[2]https://www.aminer.cn/weibo-net-tweet

Evaluation: We use Hit Ratio (HR@K) and Normalized Discounted Cumulative Gain (NDCG@K) [116] as metrics. HR@K is defined as:

$$HR@K = \frac{\#hits@K}{\#tests} \tag{3.8}$$

where $\#tests$ is the total size of testing data. In Microblog-Hashtag recommendation, it means the number of testing tweets, and in User-Hashtag recommendation, it's the number of testing users. $\#hits@K$ is the number of hashtags which were successfully recalled in the top-K hashtags ranking list. And NDCG@K is defined as:

$$NDCG@K = \frac{DCG@K}{IDCG@K} \tag{3.9}$$

where $DCG@K = \sum_{i=1}^{K} \frac{2^{rel_i}-1}{log_2(i+1)}$ to accumulate the graded relevancy rel_i of hashtags at position i. $IDCG@K$ is the $DCG@K$ score of perfect ranking list for normalization. In our experiments, we set rel_i as binary relevancy as [116] did.

The dataset was split randomly into two parts with ratio 8:2 as training set and testing set respectively. Validation set was used for tuning the hyper-parameters and later combined with training for final model. As [111] did, for each positive interaction, we sampled 99 hashtags that have no interaction with this user (for User-Hashtag recommendation) or this microblog (for Microblog-Hashtag recommendation), and evaluate the ranking list among these hashtags.

Baselines: This work is mainly focused on UHR task. As mentioned UHR solely based on graph data has not been investigated in previous works. As such, we compare the model with baselines that perform based on historical user-hashtag interaction data and textual data. The following is a brief description of the baselines were used for the UHR task.

- **Classification-Based Recommender (CBR):** Analogous to [117], we model the recommendation task as a multi-class classification problem for textual data with word frequencies as textual features.
- **LDA-Based Recommender (LDAR):** [118] employs a model based on topic modeling over textual data to recommend hashtags.
- **LSTM-Based Recommender (LSTMR):** In this method, content of microblogs are encoded using LSTM and then obtained encoding is assigned to a hashtag [114].
- **Matrix Factorization (MF):** [79] feeds user-hashtag historical interaction data to a Matrix-Factorization-based model to make hashtag recommendations for users.

- **Memory Network-Based Recommender (MNR):** [115] is a novel end-to-end memory network to model users based on the content of their microblogs. We have adopted the model for UHR task.

- **PCA-Based Recommender (PCAR)** We have developed this baseline to further evaluate our general framework that relies on profiling users based on aggregating the profiles of hub nodes. This baseline has the same structure as PHAN which feeds the embeddings of the target hashtag and user to a GMF model. However in this baseline, we use PCA to generate user embeddings. That is, we consider the connections of a user towards hub-nodes as a binary feature vector. Then apply PCA to the binary feature vector and use the output of PCA as user embeddings.

It should be noted, **CBR**, **LSTMR** and **LDAR** have been originally proposed for MHR task. We adopt them for the UHR task. In fact, instead of feeding a single microblog, a set of microblogs were fed to the models.

Moreover, to show the effectiveness of the model as a personalized recommendation model for MHR task, we compared it with text-based MHR models including **CBR**, **LSTMR**, **PCAR** and **LDAR**. Additionally, we used the following state-of-the-art text-based models as baselines:

- **CNN-Attention-Based recommender (CNNAR):** [119] employs a convolutional neural network-based architecture equipped with an attention mechanism to effectively analyze the content of microblogs for the hashtag recommendation task.

- **EmTaggeR:** [89] makes recommendations by finding the relevance of the embedding of the target microblog to that of the candidate hashtag. Embeddings of hashtags and microblogs are obtained based on the embeddings of the constituent words.

It should be noted that idea of building personalized MHR models by relying on demographic, personal and location-based information has been investigated in previous works [90, 91, 92]. However, such user features are not available in our dataset. Also, the personalization methods introduced in those works are orthogonal to PHAN.

3.4.2 Performance of the Proposed Model (RQ1 & RQ2)

The overall performance of PHAN and baselines over Twitter and Weibo datasets are displayed in Tab 3.2. Also, we evaluated four different variations of the PHAN model:

1) Supervised attention-based embedding based on co-occurance (**PHAN-SACO**), 2) Supervised attention-based embedding based on Informativeness (**PHAN-SAIN**), 3) Classic attention based embedding (**PHAN-CA**) and 4) Fixed Embedding (**PHAN-FIXD**). In PHAN-CA, we set $\lambda = 0$ in the joint objective function, i.e., the supervision for attention is not applied. In PHAN-FIXD, following the structure introduced in in [111], we fed the input followee list to the network to get the fixed embedding of the user (hashtag-independent). We used both text and graph data in all four variations.

Table 3.2: Performance of the proposed model (PHAN) and the baseline methods on the Twitter and Weibo datasets.

User-Hashtag recommendation					**Microblog-Hashtag recommendation**				
	Twitter		**Weibo**			**Twitter**		**Weibo**	
Model	**HR@10**	**NDCG@10**	**HR@10**	**NDCG@10**	**Model**	**HR@10**	**NDCG@10**	**HR@10**	**NDCG@10**
CBR	19.83%	8.43%	17.21%	7.64%	**CBR**	50.89%	24.97%	39.74%	29.12%
LDAR	18.74%	7.97%	15.96%	6.87%	**LDAR**	43.79%	22.03%	34.06%	27.43%
LSTMR	23.14%	11.59%	39.47%	27.22%	**LSTMR**	56.21%	38.31%	52.43%	35.47%
MF	26.30%	14.98%	23.26%	12.29%	**EmTaggeR**	62.47%	49.69%	57.63%	44.09%
MNR	28.29%	17.00%	30.92%	20.75%	**CNNAR**	63.64%	49.48%	55.95%	43.71%
PCAR	39.19%	23.83%	37.47%	24.91%	**PCAR**	74.84%	55.97%	61.86%	51.78%
PHAN-FIXD	41.21%	24.33%	38.43%	24.78%	**PHAN-FIXD**	74.65%	55.76%	61.58%	51.40%
PHAN-CA	41.35%	24.25%	38.46%	24.83%	**PHAN-CA**	73.46%	53.79%	61.70%	51.42%
PHAN-SACO	42.79%	26.20%	40.45%	26.48%	**PHAN-SACO**	74.64%	55.27%	63.41%	51.54%
PHAN-SAIN	**43.73**%	**26.33**%	**42.06**%	**27.94**%	**PHAN-SAIN**	**77.57**%	**58.54**%	**65.31**%	**52.94**%

1) As it can be seen in Tab. 3.2, PHAN with supervised attention outperforms the other variations. For example, in MHR task, PHAN-SAIN outperforms the PHAN-FIXD model on the Twitter dataset by 2.92% in terms of HR@10; For UHR task, we observed the same behavior whereas on the Weibo dataset PHAN-SAIN outperforms PHAN-FIXD and PHAN-CA by at least 3.5% in terms of HR@10. The results confirm the effectiveness of weakly supervised attention. In fact, while the fixed embedding model has comparable results to the classic attention model, applying the supervision significantly improves the results. Also, PHAN-SAIN provides better results than PHAN-SACO, which means that the form of weak supervision is of great importance.

2) The results strongly confirm the effectiveness of incorporating link data in hashtag recommendation for UHR. PHAN outperforms all baselines which consider only text data. For example, PHAN-SAIN outperforms an advanced text-based profiling model like MNR by 15.44% and 11.14% in terms of HR@10 on the Twitter and Weibo datasets respectively. It should be mentioned, the experiments were done only on users who have associated text data. However, a large portion of users do not generate content, hence, text-based models are not applicable to them.

3) Also for MHR, we observe that the idea of personalized recommendation based on graph data is quite effective. For example, in MHR task, PHAN-SAIN outperforms CNNAR by

12.94% in terms of HR@10 on the Twitter dataset.

3.4.3 Effectiveness of Supervised Attention (RQ2)

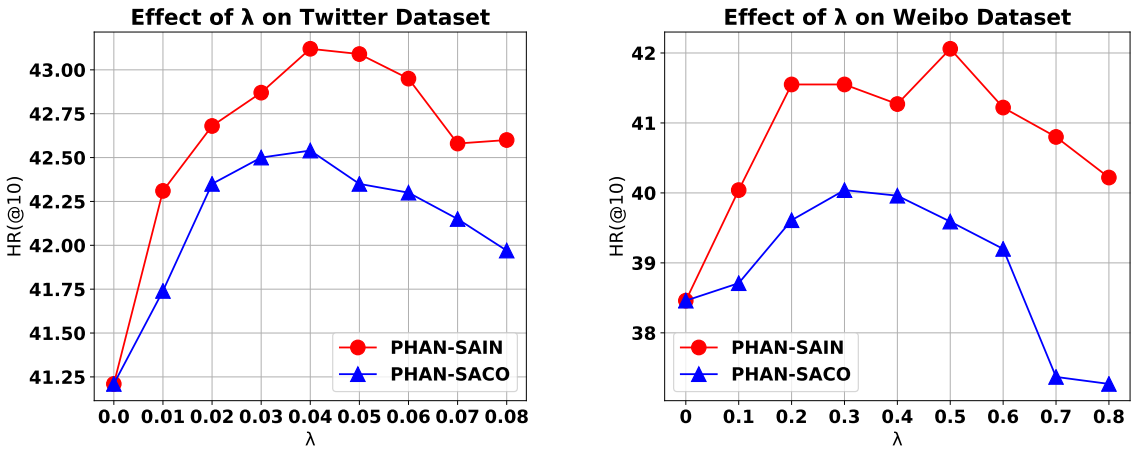

Figure 3.4: HR as a function of λ on Twitter and Weibo Datasets.

In this first section of this experiment, we investigated the model's performance as a function of λ, which determines the importance of attention supervision during training. Figure 3.4 shows the effect of increasing λ for UHR task for both of our supervision models. We observe two behaviors in these figures. Firstly, the variation trend of $HR@10$ with varying λ is quite stable across the two datasets. Initially, as we increase λ, $HR@10$ increases but later it declines. This is because higher values of λ misled the models' objective function to fit the relevancy between representative nodes and hashtags rather than to fit the relevancy between targeted users with hashtags. Secondly, the optimal λ differs for two datasets.

In the second part of the experiment, we show how supervised attention aids the model to better determine the relevance of representative nodes and hashtags. To this end, we first find the co-occurrence-based weak labels for a set of hashtag-representative node pairs and split the obtained values into multiple bins. Next, we find the relevance score of the set of the pairs within each bin by feeding their embeddings to the GMF function. The embeddings are obtained based on two versions of the model: without supervised attention a nd with co-occurrence-based supervision. The figure 3.5 shows the average relevance score of the embeddings of hashtag-representative nodes pairs as a function of weak label bins for two versions of the model. For example, for the set of pairs with weak labels within [0.0,0.2], the average relevance score obtained from embeddings with and without weak supervision are 0.12 and 0.24, respectively. As it can be seen, the relevance scores obtained from PHAN with

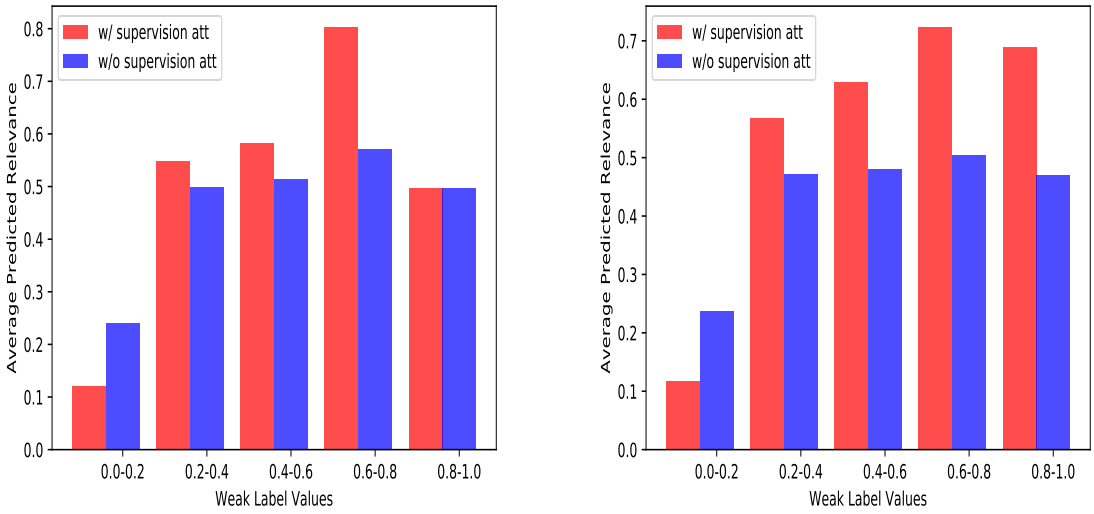

Figure 3.5: Average relevancy value between embeddings of representative nodes and hashtags with and without supervised attention for different bins of weak labels on Twitter (left) and Weibo (right) datasets.

supervision are more correlated with the weak labels than the ones obtained from PHAN without supervision. In fact, by employing the idea of supervised attention, we guide the model to find embeddings in a way that the relevance of hashtags and representative nodes become more aligned with the weak labels. Note that we do not claim that weak labels are fully accurate in determining the relevance scores; however, we suggest they can be beneficial in better recognizing the pattern, and we use the parameter λ to adjust their involvement level in the learning process.

3.4.4 Scale-Freeness of the Network (RQ3)

In this experiment, we study the validity of our basic assumption: What proportion of links are received by representative nodes? Fig. 3.6 depicts the degree distribution of the user's links towards the entire set of nodes (raw) and towards representative nodes (filtered) in our Twitter and Weibo datasets. The set of representative nodes were selected by setting L to 120 and 10 in Twitter and Weibo datasets, respectively. As it can be seen, the out-degree towards representative nodes tends to be smaller than raw out-degree; however, a large percentage of users have out-degree larger than 10 towards representative nodes in both of the datasets. In other words, the idea of user profiling based on representative nodes is effective for a large proportion of users.

It should be noted that the proposed model can be further developed to become effective

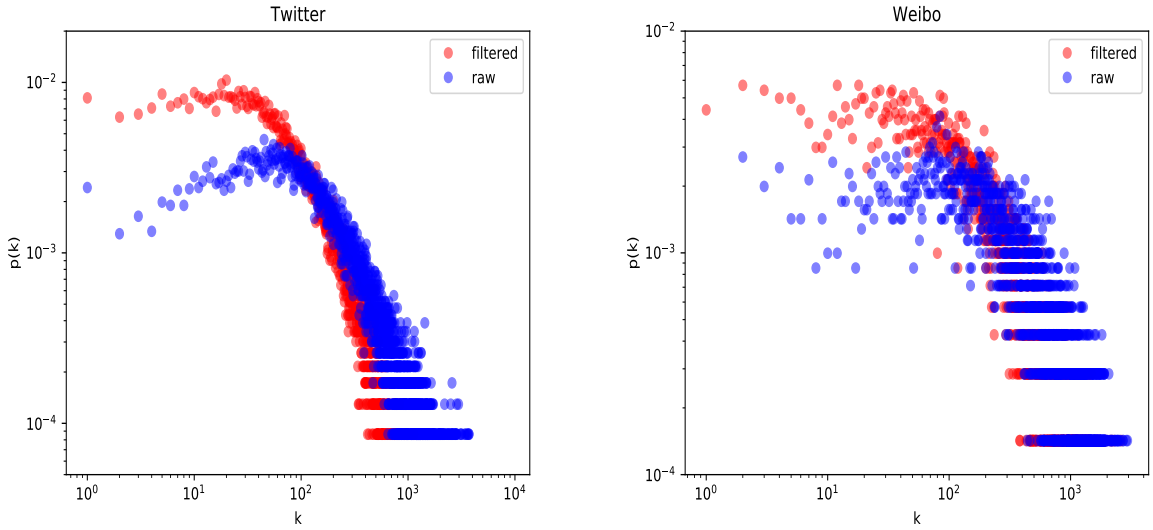

Figure 3.6: Distribution of the outdegree of users towards representative nodes (filtered) and towards entire set of nodes (raw).

even for those users who do not have a sufficient number of links towards representative nodes. For such users, as an alternative solution, we can rely on ordinary (non-representative) neighbors. That is, the target user's ordinary neighbors can be embedded based on their links towards representative nodes. Again, based on scale-free property, ordinary-neighbors are expected to have abundant links towards hub-nodes, which enables us to profile them based representative nodes. And next, the target user can be profiled based on the profiles of the ordinary neighbors. Indeed, we still rely on hub-nodes; however, we use ordinary-users as bridges to reach representative nodes.

3.5 CONCLUSIONS

In this chapter, we studied the problem of User-Hashtag Recommendation to facilitate the efficient use of hashtags on microblogging websites. Applying classic recommendation models to address the problem based on content data faces major drawbacks due to sparsity and ambiguity of content data. In fact, most of users are not active in terms of generating content/text data. Our key idea was to profile users' interest based on graph data and leverage it in the recommendation process. To efficiently exploit graph data, we introduce the notion of representative nodes and suggest that the embeddings of users in scale-free networks can be derived from these nodes which to our knowledge is the first instance to do so in graph mining context. In fact, in our proposed model, we softly cluster the network

by profiling the cluster centers of the network. Then profile users within the network with respect to the clusters they are connected to. On top of this general idea for user profiling in scale-free networks, we further propose a weakly supervised attention mechanism to perform a target/hashtag aware aggregation of representative nodes. Extensive experiments on two real-world datasets obtained from Twitter and Weibo websites verified the effectiveness of our model. Also it should be noted that while we have applied the proposed recommendation model for hashtag recommendation task, it can be viewed as a general user profiling method based on representation learning. Hence, it can be applied to different applications such as friendship recommendation [120], targeted advertisement [121], and personalized feed in social networks [122].

CHAPTER 4: GRAPH-BASED USER DISCOVERY

4.1 OVERVIEW

In this chapter, we introduce a novel interest-based user modeling problem denoted as interest-based user discovery. Discovering users with certain interests on social networks like Twitter has interesting applications. For example, a key consideration of viral marketing and targeted online advertisement is to efficiently identify groups of users sharing certain combinations of interests [123]. Another interesting application is social network-based polling systems, i.e., a polling system can be developed on social networks to identify users with certain patterns of opinions/interests to aid managers and organizations to make more informed decisions [124].

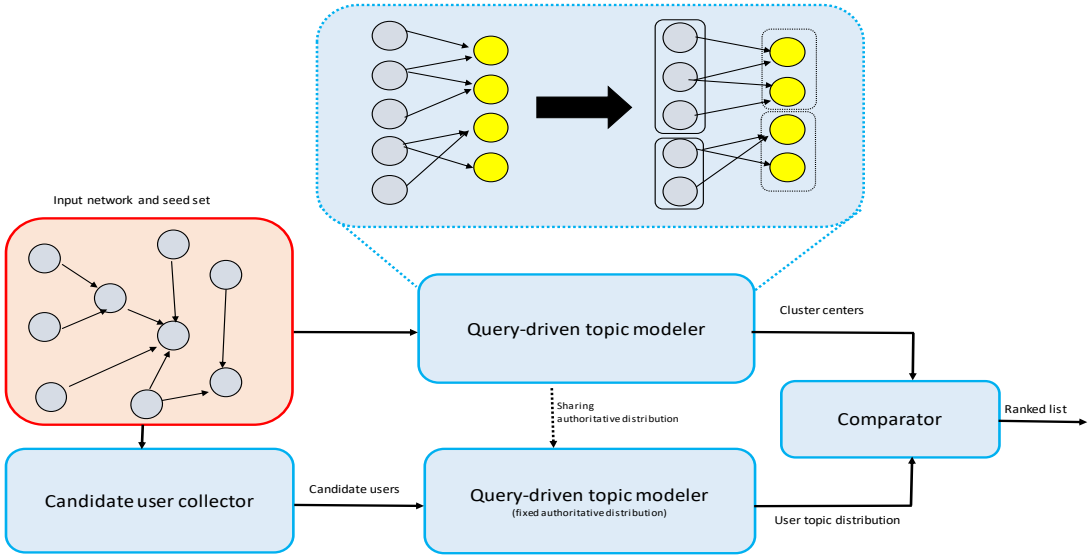

Figure 4.1: The structure of the proposed user discovery model. The query driven topic modeler identifies the structure of the query and determines the target interest combinations. The candidate collector retrieves a set of candidate user based on the notion of hub nodes. The linkage patterns of the candidate users are obtained using the topic modeler and compared to the patterns extracted from the seed set. The ranking is generated based on the comparison results.

In this thesis, for the first time, we aim to build a user discovery system on microblogging websites named as **Interest Pattern-based User Discovery** (**IPUD**): a system that can generate a ranked list of users according to their relatedness to a given interest-based query. While user interest identification has been extensively studied in previous works, interest-based user discovery is inherently different. The difference of a user discovery system from

an interest identification system is mirrored in the difference of classic document categorization methods and document retrieval models [125]. Towards building the system, two fundamental design questions should be answered. First, how input queries can be defined? Second, what data source can be leveraged to retrieve the user related to a given query?

Query definition: To comprehensively comply with user's information needs, we assume that an interest-based query may involve a request to determine users associated with one or multiple *interest combinations* where we define an interest combination as the conjunction of interests over a set of interest topics. For example, a system user may request to retrieve two groups of users A and B with different interest combinations where A is the users interested in politics and basketball and B is the users interested in politics and soccer. However, representing the notion of interest topic remains the key challenge to define a query considering the vague notion of interest topic. Classic query representation methods, e.g., those based on a predefined set of keywords, hashtags, or entities in knowledge-bases [126], have limited effectiveness because it is unrealistic to expect searchers to be aware of the entire predefined set of interests. As such, ad-hoc queries need to be created. To this end, we devise to employ the notion of *query by example*, i.e., a searcher can define the target query by picking a set of examples of users aligned with the target query denoted as a *seed set*. The idea of defining queries based on examples has been used in information retrieval systems for which formulating complex input queries is challenging [127, 127, 128].

Data source: User's interest in a social network are reflected in two major data sources: content data and graph data. However, only a small proportion of users generate content on a network like Twitter. For example, around 50% of Twitter users have never sent a tweet [129]. Moreover, text data is inherently ambiguous, noisy, and language-dependent. It has been shown that user's links also indicate user's interests [5]. Unlike content data, it is a well-structured data source and also more abundant than content data because most of the users on social networks tend to be 'listener'. As such, we opt to use graph data to build the system.

Although we aim to build our system based on graph data, to build a more comprehensive system, content data can be involved as well. In fact, for active users who generate content data, a more accurate profile of a user can be constructed by leveraging both content and graph data. Moreover, we can leverage content data to enrich queries. That is, given a seed set of users, we can determine the shared interest of the users based on the content data generated by them. Next, we can retrieve other users who share similar content. Although the retrieved users cannot cover all of the users associated with the query, it enables us to augment the seed set of users. This idea is more helpful in settings where the size of a given seed set is smaller. Despite the obvious usefulness of content data to build a user discovery

system, we define our problem solely based on graph data. This allows us to mainly focus on challenges associated with graph data. However, a model developed based on graph data can be further extended to leverage content data. In fact, the idea of fusing graph and content data can be explored in future works.

Problem definition: Considering our answers to the aforementioned design questions, the central problem towards building IPUD becomes expanding the target seed set of users to find and rank the related users by analyzing their following connections. We name this problem as interest-based seed expansion and, in short, refer to it as **seed expansion**.

Building an *efficient* model to address the problem is challenging due to the *ad-hoc* nature of the queries as well as the huge size of the Twitter network and its high complexity. The input queries of the system are not predefined and routinely processed. Hence, an offline model that performs based on profiling the entire set of users is not effective because it needs to rely on predefined sets of topics [55, 130]. As such, retrieving relevant users must be performed in an online and query aware manner. However, the network involves billions of edges and millions of nodes where edges may represent complex meanings. How can we efficiently discover the interest patterns embedded in the seed set and determine the relevance of potentially millions of candidate users to those patterns?

Analogous to the generative process of documents, it can be said that a user u is associated with a set of interest topics where u chooses to follow a user v based on how much v represents a topic of her interest; hence, we suggest that the core idea behind topic modeling [131] can be employed to determine the structure of the input query. In fact, we can view a user and her followees as a document and words of the document, respectively [132]. As such, intuitively, by analyzing the seed user's outgoing links we can determine how much each user on the network represents each of the target topics which we refer to it as *authoritative distribution*. And consequently, the topic distribution of the seed users links towards the target interest topics can be determined. This allows us to discover the structure of the query, i.e., we can identify the clusters of seed users representing each interest combination and determine how each group is associated with the target topics. Next, given candidate users, we can obtain the topic distribution of their links towards the target topics, compare them to the query structure and eventually rank them. We can also collect candidate users from the followers of the user with high authoritative values in one of the target topics since such users are more likely to be relevant ones. Fig. 4.1 represents the overall structure of the proposed discovery model based on the introduced idea. And Fig. 4.2 is a toy example depicting the idea of topic modeling over users connections. Nevertheless, simply applying a generic topic modeling to seed users outgoing connections would not lead to the desired outputs due to the challenges associated with the problem:

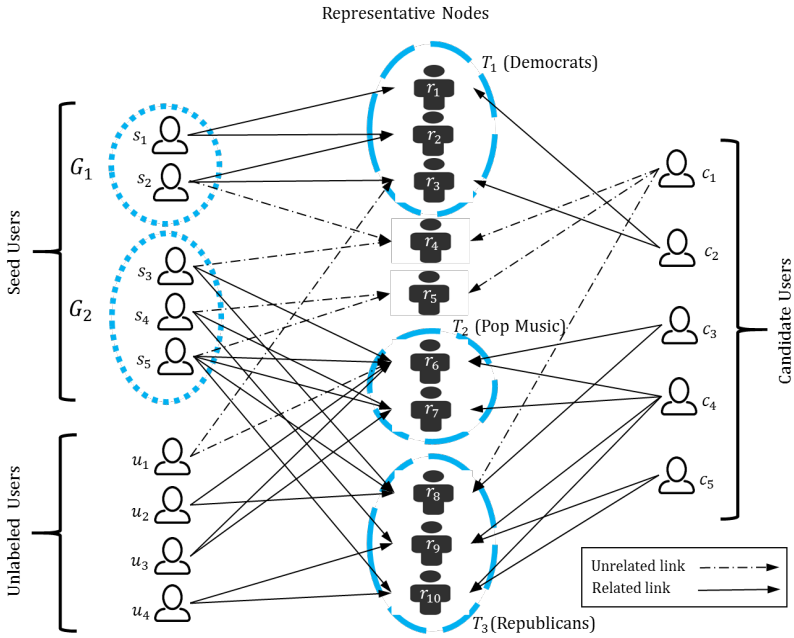

Figure 4.2: A toy example: the seed set represents a query that consists of two interest combinations: a) users interested in democrats, b) users interested in republicans and pop music.

Huge vocabulary size. Finding the authoritative distribution of the target topics based on seed users outgoing connections is challenging due to the huge size of the vocabulary. Topic modeling methods rely on co-occurrence patterns. Our vocabulary (the set of users on Twitter) involves millions of users while the size of a seed set is limited to a few hundreds of users. As such, the co-occurrence patterns are quite sparse. Hence, the authoritative distributions of the target topics cannot be obtained accurately. Consequently, the structure of the input query cannot be discovered, and additionally, we cannot collect meaningful candidate sets.

Hidden topics. In order to determine the structure of a query, it is key to identify the *target* topics. The nature of our problem requires us to do topic modeling in a way that the obtained topics provide high distinguishability between a seed user and an unrelated user, i.e., the topic distribution of a seed user should have high *distance* from that of an unrelated user. However, discovering such topics is challenging due to two main reasons:

1) Irrelevant links: Not all of the connections of seed users indicate their interests in the target topics. In other words, an outgoing link of a seed user may either indicate the user's interest in the target topics, or her interest in a topic irrelevant to the target interest pattern. In the toy example, links towards r_4 and r_5 do not indicate interest in the target topics. It is not accurate to leverage irrelevant connections to infer the target interest patterns. Now

the question is how to filter out irrelevant connections?

2) Indistinguishable topics: Distinguishing different topics using a topic modeling method requires having data points with different topic distributions over the target topics. However, if we perform topic modeling solely based on seed users, there is a possibility that we cannot discover all topics involved in the query. In fig. 4.2, there exist three seed users s_1, s_2 and s_3 interested in both T_2 and T_3. Analyzing the co-occurrence patterns of these users would not aid us to distinguish representative nodes of T_2, from those of T_3. Failing to identify the target authoritative distributions may lead to inaccurate outputs. Consider two candidate users c_3 and c_5 where c_3 is interested in both $T2$ and t_3 while c_5 is only interested in T_3. Intuitively, c_3 is more relevant to the seed set than c_5. However without accurately determining the target authoritative distributions of T_2 and T_3, distinguishing c_3 from c_5 would not be possible. However, how to accurately discover all of the interest topics involved in the target interest pattern?

Clustered patterns. A query may involve multiple interest combinations where there exist some seed users representing each of them while it is not known what interest combination a seed user represents. In our toy example, there are two groups within the seed set: G_1 and G_2. Users within G_1 are interested in T_1 while users within G_2 are interested in T_2 and T_3. While in classic topic modeling, topic distributions are defined for each individual entity, in our setting, the question becomes how to identify the groups within the seed set and subsequently determine the topic distributions associated with each group?

In response to these challenges, we propose a model denoted as **query-driven topic modeling** which addresses the aforementioned challenges in an integrated manner. Unlike generic models, it tries to identify topics in a way that it can provide distinguishability between a seed user and a random one. The proposed generative model is robust against sparsity and involves filtering irrelevant connections, clustering seed users, and discovering informative topics. More specifically, the model relies on the following ideas:

Query-specific vocabulary selection. In response to the challenges of huge vocabulary, we suggest that linkage pattern extraction can be done by only analyzing seed user's links towards nodes popular among them. We recognize popular nodes in the network as precious elements to effectively and efficiently address the problem. Twitter is a network with high scale-freeness [7, 17, 18]. It has been shown that for each topic of interest, there exists a set of hub/representative nodes, and they receive a large proportion of the links representing the topic. Hence, we propose that we can define our authoritative distributions based on representative nodes of the target topics. Statistically, we expect seed users to have a sufficient number of links towards the hub nodes of target topics. As such, given a set of users, we determine the hub nodes associate with the query by selecting the nodes popular

among seed users.

Query-focused topics. We use two techniques to guide the model to focus on identifying target topics: filtering unrelated links and leveraging unlabeled users. 1) In order to filter irrelevant links, we define a binary latent variable for each edge of seed users towards hub-nodes indicating if a link is *related*. In the link generation process, the value of the variable is determined. We define the priors of the variable based on statistics obtained from the network. 2) Moreover, we sample unlabeled with respect to the target query and exploit them in the topic modeling process to distinguish topics of right granularities. The unlabeled users are randomly selected from the representative nodes of the target topics. Back to our toy example, some of the unlabeled users are expected to be interested either in T_2 or T_3. This way, involving them in the process can aid us to discriminate T_2 and T_3. We designed a different generative process for unlabeled users than seed users because they are inherently different. However, we used parameter sharing between these two generative models.

User-Hub co-clustering. We propose to cluster seed users along with discovering topics represented by hub-nodes. To this end, a latent variable is defined to decide what cluster a user belongs to, where each cluster is associated with a topic distribution. As such, in the link generation process, the target link of the seed user is generated with respect to the selected cluster. Note that it is possible to do clustering and topic modeling separately. However, since they are related, performing them in an integrated way can improve both clustering and topic modeling.

Our experiments were conducted on datasets obtained from Twitter. We collected seed sets and a candidate set for each seed set containing both related and unrelated users. The quality of the ranking outputs was measured using hit-ratio and nDCG. The results show the model can effectively rank relevant users superior to irrelevant ones and outperform the baselines.

4.2 PROBLEM DEFINITION AND PRELIMINARIES

We first formally introduce the problem and then briefly explain the link generation process on Twitter and the idea of topic modeling over graph data for user-interest identification.

Problem formulation. Here we provide a formal development of the seed expansion problem. Let $G = (V, E)$ denote the directed followee/follower graph on Twitter where V is the set of users with size $|V|$ and E is the set of edges $e(u, v) \in E$ within G with size $|E|$ in which $u \in V$ and $v \in V$, indicating that u follows v. Assume we are given G as well as a seed set of users $S \subset V$ and the task is to identify a ranked list of users over G according to their relatedness to S. We assume that the seed set represents an interest pattern which is

defined as a set of multiple interest combinations where an interest combination indicates a combination of interests in a set of topics, e.g., ((topic A AND topic B) OR topic C) is an interest pattern with two interest combinations. Note that efficiency is the key requirement of the problem since the purpose of solving the problem is to build a discovery system.

Link generation process on Twitter network. Two major reasons can be considered for the creation of a link for an initiator node u to a receiver node v on Twitter: 1) Social bond: there exist a social bond-based relation between u and v, e.g., friends, family, colleagues. [5, 7, 124]. 2) Interest: v is authoritative in a topic of interest that in which u has shown interest, i.e., the link can be interpreted as the user u expresses her interest in tweets written by the user v. As such, intuitively interest-based links can be exploited for interest profiling.

It can be said that interest plays a role in the establishment of the relation as the topic does in the document generation process [133]. As such, document generation models can be adapted to analyze the link generation process on Twitter: we can assume each user is associated with a distribution over latent interest topics and each interest topic is modeled as a distribution over users (authoritative distribution). This way, to generate an interest-based link, we can assume an initiator user first chooses an interest and then based on the chosen interest, the user chooses a receiver node to follow. In other words, a document in a corpus becomes an initiator user's following list, and a word becomes a receiver node in the list. This suggests that by analyzing Twitter followee edges using LDA two estimates can be delivered: the distributions of user's interests and authoritativeness distribution of users in each topic of interest. As mentioned, several works have tried to apply this general idea to identify users interests on microblogging websites [134].

4.3 RELATED WORKS

From a technical perspective, our model is related to Bayesian graph-based user interest identification techniques. And from a problem definition point of view, it is related to local community detection models.

User interest identification. Although the problem of interest-based user discovery is fundamentally different from the task of user interest identification, they are conceptually related. Several attempts have been made on profiling user's interests, in general, on social networks [135] and in particular on Twitter based on different data sources [3], including those that solely rely on graph data [136, 137]. In fact, graph-based models are an important class of profiling methods that mainly employ Bayesian models to discover latent interest-based groups on Twitter [132, 134, 138]. For example, Henderson et al. employed LDA-based

models to group users on Twitter based on their topics of interest and label their relationships [133]. Young cha et al. attempted to modify the LDA model to mitigate the problems caused by high popularity nodes on Twitter because the classic LDA model assumes that every node in the graph has roughly equal popularity [132]. Following the same line of research, Barbieri et al. built a topic modeling-based user recommender system by labeling user's connections on Twitter [5]. While in this line of work the goal is labeling links in the entire network into a predefined set of latent classes, our respective focus is on identifying the shared interest patterns of a seed user by analyzing their immediate neighbors.

Also, a class of user profiling models attempted to profile users transitively based on their popular followees. In [129, 139], the notion of lists has been used to identify the profile of popular nodes and then transitively profile other users. Lists are an organizational feature by which users can group experts on topics that interest them. Hence, names and descriptions of lists can be used to discover the topical expertise of popular users on Twitter. Analogously in [140], the Wikipedia pages of popular nodes have been analyzed to profile the topics attributed to them and next identify the interests of their followers. Moreover, recently the notion of hub-nodes has been exploited to build a hashtag-recommendation model on Twitter [21]. While these works aim to use a predefined set of hub-nodes for user-profiling, we select and analyze hub-nodes in a query-driven manner to discover the structure of a given query and we do not rely on external resources to profile hub-nodes.

Local community detection. The problem setting of interest-aware seed expansion has some similarities to that of local community detection [141]. In local community detection, the task is to find a community around a seed set of users where a community is specified as a densely connected subgraph in the network [142, 143]. However, interest-based seed expansion is fundamentally different from density-based seed expansion. While local community detection models assume that the seed users come from the densely connected clusters [144], our assumption is that seed users reflect an interest pattern. In fact, users representing an interest pattern do not necessarily belong to the same densely connected cluster.

4.4 PROPOSED MODEL: INTEREST PATTERN-BASED USER DISCOVERY (IPUD)

As mentioned, considering the nature of the problem, the patterns embedded in the input seed set should be extracted in an online manner. However, designing an efficient online model is challenging. For example, one may suggest that recent profile propagation-based models like graph-convolutional networks cannot be adopted to the problem, i.e., by 'propagating' the profiles of seed users in the network other relevant users can be discovered [80, 130]. However, the effectiveness of such models is limited particularly due to the huge

size of the network. Our core idea to approach the problem has a basis in the generative process of links. Intuitively, if we manage to identify what topic a seed user's link represent, we can determine the structure of the query. That is, we can group seed users based on the interest combinations they represent and model each group based on the distribution of their links towards the target topics. The obtained distributions can reveal the query structure. Eventually, given a candidate user, its linkage patterns can be determined and compared with the ones discovered from the seed users.

To develop a model based on the proposed idea, we rely on topic modeling of seed users outgoing links. However, a generic topic modeling is not applicable in our problem due to the challenges described in the introduction section. In the following, we describe IPUD in three steps which addresses the introduced challenges.

4.4.1 Vocabulary Selection

In response to the challenge of vocabulary size, we introduce the idea of **query-aware vocabulary selection**, i.e., we propose to focus on analyzing the generative process of seed user connections towards *hub nodes* rather than analyzing all of their outgoing connections. We propose to define the authoritative distribution of the target topics based on their hub-nodes. It has been shown that Twitter is a scale-free network, i.e, they contain some hub nodes and a large portion of links are towards hub nodes [18]. A user can receive only a limited number of social tie-based connections. This implies that when a node becomes popular/hub in a microblogging network, it boldly represents a topic of interest. In other words, there exist some popular hub nodes for each topic of interest, and if a user is interested in a topic, she very likely follows some of the corresponding representative nodes [18]. For example, a user like "Barak Obama" appears very often in the followee lists of the users interested in the USA politics. Also it should be considered almost all of the links towards hub nodes are interest-based links. Hence, it can be said authoritative distribution of the nodes also follow the power-law distribution, i.e., while the number of hub nodes is extremely smaller than the number of nodes in the network, they receive a very large portion of the links representing the target topics [17].

Hence, an important characteristic of a hub-node associated with the target topics is its popularity among seed users. In other words, we expect that representative nodes associated with the interest topics involved in the target interest pattern to appear more or less frequently among the immediate neighbors of the seed users, i.e., if a node is not popular among seed users, it is not a representative node for any of the target topics of interest and also it cannot be used as 'index' to reach relevant users to the query. We rely on this basic

characteristic to select the target set of hub nodes denoted as representative nodes. In more detail, as the initial step of the model, the set of representative nodes is selected based on their popularity among seed users. Given the target set of users U and a graph $G = (V, E)$, we define representative nodes $RN = \{r_1, r_2, ..., r_n\}$ as the set of nodes in V with at least L followers in U where n is the number of representative nodes. Accordingly, the set of representative followees of the user u is represented as $RN_u \in RN$. It should be noted, L is a parameter of the proposed model to be tuned according to desired performance and efficiency.

We are aware that by limiting our vocabulary to representative nodes, the model loses some informative signals for topic modeling. However, it can be shown that the error of filtering out non-representative nodes is more pronounced if the number of topics is larger and also less number of words within the documents belong to representative words. However, in our problem the number of topics is small and also it has been shown that a large proportion of links is towards representative nodes. Moreover, it should be considered a large proportion of links towards non-representative nodes are social tie-based links.

4.4.2 Linkage Pattern Extraction

We propose a topic modeling-based generative model for seed users connections towards the selected hub-nodes which addresses the introduced challenges. We involve two main ideas in this generative models. In response to the challenge of hidden topics, we propose the idea of **query-focused topics**. That is, we equip our model with components that allows us to first filter irrelevant connections and second involve unlabeled users in the topic modeling process. Moreover, we propose the idea of **co-clustering seed users and hub nodes** to address the challenge of clustered patterns. That is, the model not only determines how each hub node represents each of the topics (soft clustering of hub nodes) but also it clusters seed users (hard clustering) based on how they are connected to the target hub nodes in an integrated manner. In the following, we describe how these ideas are embedded.

In the proposed generative model, consistent with the problem definition, we assume there are G latent interest combinations and K latent interest topics embedded within the seed set. Each of the topics k is modeled as a Multinomial distribution ϕ_k, denoted as authoritative distribution, over RN. And each interest combination is modeled as a Multinomial distribution θ_g over the latent topics. Also, a Multinomial distribution ξ is defined over the G latent interest patterns where the distribution represents the likelihood of a seed user posing each of the G interest combinations.

Given a user u, first, the user is assigned to one of the G interest combinations. In

actuality, we model the target interest pattern with a set of groups within the seed set where each of the groups represents an interest combination. Next, to generate each of her connections, in response to the challenge of 'irrelevant connections', we first determine whether it is relevant to the interest combination assigned to the user or not. As mentioned, a seed user may have connections representing her interests in topics other than the target topics. Such connections are independent of the connections relevant to the target interest pattern. Therefore, we use two separated paths to generate them. To determine whether a link from u to v is relevant to the target interest patterns or not we take advantage of two intuitions where the first intuition is relevant to the receiver node v, while the second intuition is concerned with the initiator node u.

1) The link is more likely to be a relevant connection if there is a larger positive bias between the popularity of v among the seed users, and v's expected popularity among a set of users randomly selected from Twitter. In fact, the larger bias indicates that v has higher *informativeness* in representing the target interest pattern and accordingly a connection toward the node is more likely to be relevant to the target pattern.

2) The link is more likely to be a relevant connection if the proportion of u's links towards RN is higher (compared to the u's links towards $V \setminus RN$). The main logic behind this intuition is that the more connections u has towards RN, the more likely she is interested in some of the target topics of interest, and consequently the more likely the outgoing connections of u are relevant to the target interest topics.

We inject these intuitions into our generative model through defining priors, i.e., for each user u from a beta distribution with scaling parameter ρ_u , a Bernoulli distribution ω_u is sampled where ρ_u is defined based on the number of links u has towards RN and $V \setminus RN$. Indeed, in accordance with our intuition, ω_u shows how likely u creates a link relevant to the target interest combination. Moreover, a single Multinomial distribution φ' over RN is defined which is generated by sampling from a Dirichlet distribution parameterized by β' in which β' is defined proportional to the popularity of the nodes within RN on Twitter based on the intuition that the higher the popularity of a representative node, more likely the connection toward the node is of type irrelevant.

Based on the introduced components, to generate an outgoing connection of user u, $e_{u,i}$, we first pick a binary variable $z'_{u,i}$ by sampling from ω_u. This variable indicates the relevance of the connection. If $z'_{u,i} = 0$, we select a node from the authoritative distribution of the popularity based background topic φ'. This branch of link generation, creates 'irrelevant' connections. If $z'_{u,i} = 1$, we pick a topic $z_{u,i}$ from Θ_{η_u} and the target node is selected from $\phi_{z_{u,i}}$.

However, this generative process still cannot address the challenge of 'indistinguishable

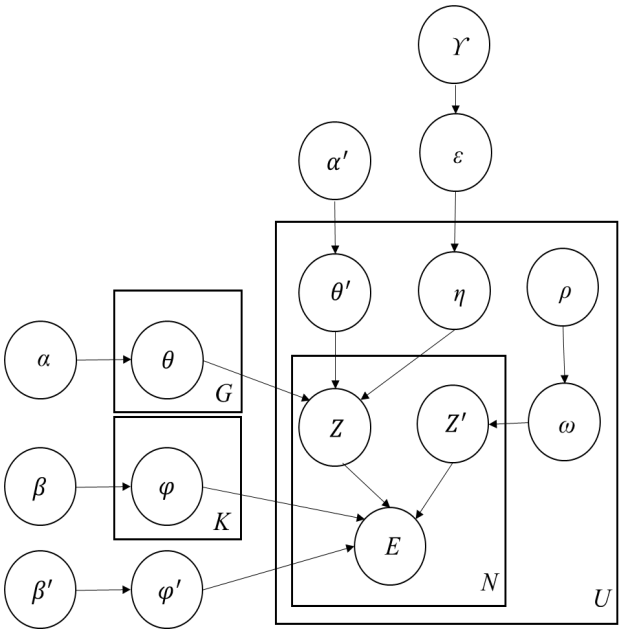

Figure 4.3: Graphical representation of the proposed generative model for linkage pattern extraction

topics'. To address the challenge, we equip the generative model with a component that allows us to augment the seed set by adding some *unlabeled users* to the pattern extraction process. In fact, the interest combinations of unlabeled users over the target topics may differ from that of the seed users. Hence, involving them in the pattern extraction process can aid the model to better identify the latent topics. However, it is important how to select unlabeled users. We sample unlabeled users in a way aware of the input query. That is, we collect unlabeled users by randomly selecting from the followers of RN.

However, leveraging unlabeled users in our pattern extraction process is challenging. An unlabeled user does not necessarily belong to any of the target interest combinations. Hence, it is not accurate to generate the links of an unlabeled user the same way that we generate the links of a seed user. To tackle this challenge, instead of assigning an unlabeled user to one of the G interest patterns, we generate an interest pattern specific to the user. In more details, if a user is an unlabeled user, to generate the connections of the user, a topic distribution θ'_u is generated by sampling from a Dirichlet distribution parameterized by α'. The rest of the link creation process is the same for the seed users and unlabeled users.

Link generation process summary: The plate diagram of the proposed generative model is depicted in Fig. 4.3. It should be mentioned, a collapsed Gibbs sampler is used to infer the parameters of the model. In summary, the process can be described as follows:

1. Draw a Multinomial topic distribution for each interest combination $\Theta_g \sim Dir(\alpha)$

where $g \in 1, 2, \ldots, G$.

2. Draw a Multinomial distribution for each topic of interest $\varphi_k \sim Dir(\beta)$ where $k \in 1, 2, \ldots, K$.

3. Draw a Multinomial distribution for the background topic $\varphi' \sim Dir(\beta')$ where β' is a vector proportional to the popularity of the nodes within RN.

4. Draw a Bernouli distribution $\omega_u \sim Beta(\rho_u)$ for each user u where ρ_u is proportional to the number of connections of u towards nodes within RN and $V \setminus RN$

5. if u belongs to the seed set:

   - Draw an interest combination $\eta_u \sim Multi(\varepsilon)$ where $\varepsilon \sim Dir(\Upsilon)$

6. if u is an unlabeled user

   - Draw a Multinomial interest combination $\theta'_u \sim Dir(\alpha')$

7. For each link initiated by user u:

   - Draw a binary indicator $z'_{u,i} \sim Binomial(\omega_u)$
   - if $z'_{u,i} = 1$
     - if u belongs to the seed set
       - Draw a specific topic $z_{u,i} \sim Multi(\Theta_{\eta_u})$
     - if u is an unlabeled user
       - Draw a specific topic $z_{u,i} \sim Multi(\theta'_u)$
     - Draw a node to follow $e_{u,i} \sim Multi(\varphi_{z_{u,i}})$
   - if $z'_{u,i} = 0$
     - Draw a node to follow $e_{u,i} \sim Multi(\varphi')$

4.4.3 Retrieving Relevant Users

Having the interest combinations for each of the clusters within the seed set as well as the authoritative distributions of the target topics, now the question becomes how to generate the candidate set and compare their interest combinations with that of the identified clusters.

We introduced two methods for candidate generation based on representative nodes: popularity and informativeness. In the popularity-based model, after selecting representative

nodes, we collect candidate users by sampling from representative node's followers proportional to their popularity among seed users. However, in the informativeness-based model, we sample the followers of hub nodes proportional to the hub node's informativeness. We define the informativeness of a hub node as the bias between the likelihood of a connection from a seed user toward the node and a connection from a non-seed user. In general, between two candidate collector, the superior model is the one that: 1) covers a larger proportion of relevant users, and 2) retrieve a lower number of irrelevant users. While the popularity-based model aims to cover a larger proportion of relevant users, the informativeness model tries to include less number of irrelevant users among candidates. This way of creating candidate sets implies that if a user has no links towards the representative nodes, she is not relevant to any of the target interest combinations.

Having the candidate set of users, their connections are fed into the generative model learned from the seed set. That is, using the generative model represented in Fig. 4.3, we can freeze the authoritative distributions of the target topics as well as the background topic, and find the interest combination over the target topics for each of the candidate user. Note that, similar to unlabeled users, we find individual interest combinations for each of the candidate users. By identifying the interest combination of the candidate users, their relevance to the seed set can be measured. To this end, using a vector distance function such as Kullback–Leibler divergence, we can find the distance of the interest combination of a candidate user from each of the interest combinations embedded in the seed set. Next, the shortest distance to the target interest combinations is considered as the distance of the user from the seed set. The reason for this strategy is that the input query may contain clusters with quite different interest combinations, and if a user has proximity to one of the clusters, she can be regarded as a relevant user. Finally, the candidate users can be ranked according to their distance from the input query.

4.5 MODEL GENERALIZATION

4.5.1 Query-driven Topic Modeling As a Method for Topic Modeling-Based Retrieval

The idea of involving topic modeling has been explored in previous works [145, 146, 147]. For example, given a set of keywords as a query, previous works have used topic modeling methods to expand queries or build query-aware language models of a target corpus. However, in this work, we employ the idea of topic modeling for information retrieval in a different setting, i.e., we use the notion of query by example to define our queries, and directly perform topic modeling on a target query. In other words, unlike previous works

that perform topic modeling on a target corpus, we apply it to a target query. As such, it can be said that in this work, we have introduced a novel approach to involve topic modeling in information retrieval.

We suggest that the idea can be applied to document retrieval context as well. To this end, first, a target query should be defined based on a set of relevant documents. Note that even if a target query is defined as keywords, we can retrieve a set of highly relevant documents using a classic keyword-based retrieval technique. Next, the retrieved documents can be regarded as 'examples' representing the query. This way of generating examples relies on the idea of pseudo relevance feedback [148]. Having the set of documents representing a query, we can perform query-driven topic modeling on the given set of documents to determine shared topics involved in the query, and retrieve documents with similar distributions over the discovered topics. As mentioned, the idea of query-driven topic modeling has some characteristics that accurately determine the intention of the system's users. It allows to find the target topics in the right granularity, and it only focuses on the target topics and filters irrelevant topics. However, this could not be achieved if we perform topic modeling on an entire corpus because such an approach is blind to a target query. It is worth noting that the proposed model for graph data cannot be directly applied to text data. Unlike graph data where a node can appear only once in the connections of a user, in text data, the number of occurrences of a word in a document could be larger than one. Hence, this difference should be considered when developing a model for text retrieval based on query-driven topic modeling.

4.5.2 Hub Node Modeling As a Clustering Technique

Analogous to the previous chapter, the main idea of the proposed technique for interest-based user discovery relies on user profiling based on hub nodes. In this section, we further discuss this idea and generalize it as an *interest-based clustering* technique.

As mentioned, for each topic of interest on microblogging networks, there exists a set of hub nodes that represents the topic. These hub nodes can be viewed as cluster centers, i.e., users interested in a topic tend to follow the hub nodes representing the topic. This implies that if we somehow determine the profile of hub nodes in the network (by determining what topics they represent and identifying their neighboring hub nodes), we can discover the interest-based clusters in microblogging networks. This idea can be developed to build an efficient technique for clustering networks. That is, having the profiles of hub nodes identified as cluster centers, we can determine what clusters a user belongs to.

Of course, the proposed idea for graph clustering is fundamentally different from the

existing graph clustering techniques that rely on this principle that clusters are defined as densely connected sets of nodes [67, 149, 150]. The classic graph clustering techniques aim to identify clusters with high clustering coefficients. However, our definition of interest-based clusters is different. We propose that an **interest-based cluster** can be defined as nodes/users that share links towards hub nodes representing the interest topic. With this definition, users within an interest-based cluster are not necessarily densely connected to each other. As such, a high clustering coefficient among users sharing a certain topic of interest is not necessarily expected. Consequently, a classic clustering algorithm cannot effectively discover interest-based clusters.

Also, it should be mentioned not only the proposed idea for interest-based clustering of the network is more effective than the classic counterparts in discovering interest-based clusters, but also it is more efficient. The proposed idea performs based on modeling a very small proportion of the nodes in the network, and clusters the rest of the nodes transitively. However, a classic clustering algorithm solves a complex optimization problem based on the clustering coefficient of clusters which needs to take into account all of the connections within a network. The idea of building an interest-based clustering technique using the notion of hub nodes can be further investigated in future works.

4.6 EXPERIMENTS

We conducted experiments to verify the effectiveness of our proposed model. We focus on answering three key questions:

- **RQ1:** How does our model perform compared to the baseline methods?
- **RQ2:** How does the model perform as a function of the seed size?
- **RQ3:** Is the model computationally efficient?

4.6.1 Test Methodology

The IPUS model can be viewed as a retrieval system. As mentioned, the proposed discovery model involves two components: a candidate generation model and a relevance prediction model. The candidate generation model is a fast method to retrieve potentially relevant users while the second system is a more accurate system with a higher computational cost. We evaluate the effectiveness of both models. Classically, a relevance prediction model is assessed by determining the relevance of the results to the query. We follow the same method

to evaluate the relevance prediction component. We create a seed set and a pool of candidate users for each of our target queries. The sets of candidate users contain both relevant and irrelevant users. After training the model based on the seed set, the users within the pool are fed to the model, and ranked according to their distance from the seed set. The performance of the model can be evaluated in terms of the model's effectiveness in ranking relevant candidate users superior to the irrelevant ones. Additionally, we evaluate how our hub node-based candidate generation model performs. To this end, we evaluated how accurately the model covers relevant users.

Dataset: We experimented with four real-world datasets collected from Twitter. Each dataset involves a seed set (representing a query) and a set of candidate users containing both relevant and irrelevant users. We relied on content data to collect our Twitter datasets. That is, we assume that a user's tweet about a certain topic can be considered as a reflection of her interest in the target topic. To this end, we used a dataset that contains user's tweets [151], and employed two approaches to identify user's interests from their tweets: keyword-based and topic modeling-based methods. The reason for using two different approaches to collect datasets is that keyword-based topics depict a narrow definition of topics while topic modeling-based method provides a broader definition. In the keyword-based approach, we defined a topic based on a set of keywords, and sampled users from our original dataset with tweets containing those keywords as relevant users. In the topic modeling-based approach, 100000 users were sampled from the dataset along with their tweets. A tweet topic modeling algorithm was applied to the user's tweets and 100 topics were obtained (each tweet was assigned to one of the topics). Based on the topics assigned to the user's tweets, a topic distribution of user's interests over the target topics were obtained, which was used to collect users relevant to a certain interest topic.

Also for each dataset, a set of irrelevant users were sampled. We purposefully aimed to draw irrelevant users among the users that have some proximity to the query. For example, given a query like 'users interested in a soccer club', irrelevant users were selected among the ones who have exhibited some interest in soccer. This way of collecting candidate pools permits us to better evaluate the effectiveness of the model. In fact, given a query, a large proportion of the irrelevant users can be easily filtered out even using a simple heuristic model. Having the user id of the set of relevant and irrelevant users for each dataset, their followee lists were obtained using the Twitter API. In the following, each of the collected datasets is introduced. K1 and K2 are keyword-based datasets while T1 and T2 are topic-based datasets.

K1: Democrats. The target query for this dataset is the users interested in the Democratic party. The set of irrelevant users were selected among users interested in the republican party.

K2: NFL And NBA. The query of this dataset is the users interested in both NBA and NFL. The irrelevant users for this query were selected among users who had shown interest only in one of the topics.

T1: Music And Technology. Based on the topics obtained from topic modeling over the tweets, we defined a query based on two topics about technology and music. We selected the irrelevant users among users only interested in one of the topics.

T2: Sports And Technology. The query of this dataset is the users interested in both sports and technology topics. The irrelevant users for this query were selected among users who had shown interest only in one of the topics.

Table 4.1: Performance of IPUD and the baselines

	K1		**K2**		**T1**		**T2**	
Model	**HR**	**nDCG**	**HR**	**nDCG**	**HR**	**nDCG**	**HR**	**nDCG**
GAN	72.28%	76.32%	49.74%	55.41%	**31.18%**	**36.87%**	53.19%	52.43%
SM	33.86%	31.76%	22.67%	26.59%	27.90%	31.52%	31.22%	33.51%
BM	62.48%	65.73%	33.70%	39.64%	26.50%	30.49%	49.07%	51.64%
CM	31.96%	34.20%	26.41%	28.19%	23.78%	25.73%	27.13%	28.77%
IPUD	**90.82%**	**91.96%**	**72.40%**	**67.42%**	21.65%	29.63%	**54.13%**	**58.09%**

Metrics: Hit Ratio (HR@K) and Normalized Discounted Cumulative Gain (NDCG@K) [116] are two major metrics to evaluate the relevance prediction model. We used them in our experiments. HR@K is defined as:

$$HR@K = \frac{\#hits@k}{k} \tag{4.1}$$

where $\#hits@K$ is the number of users which were successfully recalled in the top-K users ranking list. We set k to the number of positive examples in our test sets. And NDCG@K is defined as:

$$NDCG@K = \frac{DCG@K}{IDCG@K} \tag{4.2}$$

where $DCG@K = \sum_{i=1}^{K} \frac{2^{rel_i}-1}{log_2(i+1)}$ to accumulate the graded relevancy rel_i of users at position i. $IDCG@K$ is the $DCG@K$ score of perfect ranking list for normalization. In our experiments, we set rel_i as binary relevancy as [116] did.

4.6.2 Performance of Relevance Prediction (RQ1)

In this experiment, we evaluate the relevance prediction component of the proposed model. The size of the candidate set for each dataset is 3000 users with 500 relevant users. And the

size of seed set is 800 users (along with 800 unlabeled users). We developed four different baselines that determine the relevance of a candidate user to a seed given users links towards hub-nodes. In fact, these methods still rely on our core idea of relevance prediction based on the notion of hub-nodes. However, we use them as baselines to extract meaningful patterns from seed users connections and compare them to that of a candidate user.

Note that we have used classic clustering-based techniques such as classic seed expansion techniques as baselines because they do not fit to the setting of the problem. In fact, apart from the fact that such techniques have been designed to discover users sharing similar interest, they face efficiency challenge and need to observe the entire network. Consider an existing seed expansion technique that performs based on the notion of local clustering. The model aims to discover a local cluster of *densely* connected nodes that seed users belong to. This key assumption does not hold in our setting because users relevant to a query (a given seed set) are not tightly connected. Moreover, to discover local clusters of a seed set of users, all of their immediate and higher order neighbors need to be crawled and observed which is not feasible in our problem setting. Lastly, even if we manage to fully observe the network, the process of analyzing the network to discover the local clusters of seed users is computationally expensive and does not adhere to the efficiency requirement of the problem. As such, we define/design our baselines as different classes of methods that rely on the notion of hub nodes.

Generative Adversarial Network (GAN): This baseline can be viewed as a representative of deep generative models that are used for relevance prediction [152, 153]. We developed an adversarial learning-based model that involves generative and discriminator components. The generator generates the connections of a seed user, i.e., the model determines what hub-nodes a user is connected to. And given the connections of a user, the discriminator model determines if the connections belong to the 'real' seed users or represent a 'fake user' generated by the generative model. After training this model, we can use it to decide if a candidate user is a relevant user or an irrelevant one.

Shallow Embedding (SE): Network embedding-based models can be used to 'profile' a cluster of users in the network and then find nodes with similar profiles [154, 155]. In this baseline, we use this methodology to build the relevance prediction model. Having the seed set of users as well as unlabeled users, we build a bipartite network of users and hub nodes. Next, an embedding technique like Node2vec is applied to the network to embed the nodes [12]. Having the embeddings of hub-nodes, the embedding of a seed user is defined as the mean of the embeddings of the hub nodes the user is connected to and the embedding vectors of seed users are clustered into K clusters. Given a candidate user, the embedding of the users is computed by averaging the embeddings of the corresponding hub nodes. Finally, the

relevance of the user to the seed set is determined by finding the distance of its embedding to that of the cluster centers. Note that different embedding techniques can be used to embed the bipartite network, however, we use node2vec as a representative of shallow embedding models.

Bayesian Model (BM): We applied a Naive Bayes model to determine the relevance of users, i.e., the selected hub-nodes and seed users are considered as features and training data respectively. The relevance score of a candidate user is obtained by applying the naive Bayes model on its connections towards the hub nodes.

Counting Model (CM): The model defines the relevance of a candidate user as the number of its links towards the hub nodes. The purpose of using this simple model as a baseline is to show the importance of extracting complex patterns from user's connections to build an effective relevance prediction model.

Tab. 4.1 represents the HR and nDCG of the methods on four datasets. As it can be seen, the HR/nDCG of the proposed model is higher than the baseline models on all of the datasets except for T2. The higher HR and nDCG of the model can be linked to the power of the model in extracting the linkage patterns of the seed users based on the 'groups' of hub nodes, leveraging unlabeled users in the process and filtering irrelevant connections. For example, simple models like CB and BM do not distinguish different clusters of hub nodes. As a result, they deliver low performance on our datasets, in particular on complex queries. Also, the baseline models are not equipped with systematic techniques to leverage unlabeled users and filter irrelevant connections. The proposed model significantly outperforms the HR of a powerful model like GAN on K1 and K2 datasets by 18.44 %and 15.64 % respectively. It is worth mentioning that we conjecture the reason for the lower performance of IPUD on T1 and T2 compared to K1 and K2 is that the topics involved in T1 and T2 are broader than topics of K1 and K2. As a result, they need to be covered by a larger number of hub nodes. Hence, on smaller seed sets the performance gap between different models becomes less visible. In the next subsection, we show that the model's performance on T1 and T2 improves when we increase the size of the seed set.

4.6.3 Effectiveness of the Different Components of IPUD (RQ1)

The proposed model is equipped with different components including clustering, filtering irrelevant connections, and incorporating unlabeled users. In this section, we evaluate each of them separately. To this end, three different variants of the model were developed:

IPUD Without Unlabeled Users (IPUD-WU): In this variant of IPUD, we only feed seed users to the model, i.e., unlabeled users are not involved in the topic modeling

process.

IPUD Without Filtering Irrelevant Links (IPUD-WF): In IPUD, we define a latent variable to determine if a connection is relevant or irrelevant. However, in IPUD-WF, we assume all of the links are relevant and represent one of the target topics.

IPUD Without Clustering Component (IPUD-WCC): In IPUD-WCC, we remove the clustering component from the model and obtain the topic distribution for each individual user. The relevance of a user to a seed set is defined based on its 'distance' from the closest seed user.

Tab. 4.2 depicts the HR/nDCG of the different variants of IPUD. The setting of the experiment is the same as the previous one. As the result show, IPUD outperforms the other variants of the model (except for HR on T1). For all of the datasets, the HR of the model is higher when unlabeled data was used to build the model compared to when only the seed set was used. Indeed, the obtained results are consistent with our intuition that the diversity of interest combinations for users used in the learning phase is of great importance to extract informative topics. Additionally, IPUD outperforms IPUD-WF which confirms that eliminating irrelevant connections can improve the pattern extraction process. Moreover, it can be said that IPUD-WCC is more vulnerable to the problem caused by noisy seed users because it compares candidate users to individual seed users.

Table 4.2: Performance of the different variants of IPUD.

	K1		**K2**		**T1**		**T2**	
Model	**HR**	**nDCG**	**HR**	**nDCG**	**HR**	**nDCG**	**HR**	**nDCG**
IPUD-WU	59.17%	61.39%	44.91%	45.74%	23.25%	23.84%	47.60%	46.62%
IPUD-WF	62.48%	64.70%	33.70%	35.52%	**29.54%**	29.49%	48.19%	50.30%
IPUD-WCC	36.73%	41.55%	28.68%	28.03%	16.91%	18.57%	19.92%	17.40%
IPUD	**90.82%**	**91.96%**	**72.40%**	**67.42%**	21.65%	**29.63%**	**54.13%**	**58.09%**

4.6.4 Performance of the Model As a Function of the Seed Size (RQ2)

One of the important settings of the proposed model is the size of the seed set. Clearly, we are interested to build our model based on fewer numbers of seed users because for some queries manually collecting a substantial number of noiseless seed users could be quite challenging. Fig. 4.5 shows the precision of the model over the datasets as a function of the seed set size. According to the figure, roughly speaking, for all of the datasets at the beginning the precision increases by adding more users to the seed set, and once the seed reaches to a certain size, the precision remains the same by adding more seed users. Also, it should be noted that for the smaller sizes of seed sets the precision of the model is not

quite stable. That is, in different runs of the model on a seed set with two hundred users, we may get HRs with high variance. However, the variance is lower for larger seed sets. Additionally, we observe that as the complexity of a query increases, more seed users are needed to get high/stable HR and nDCG results.

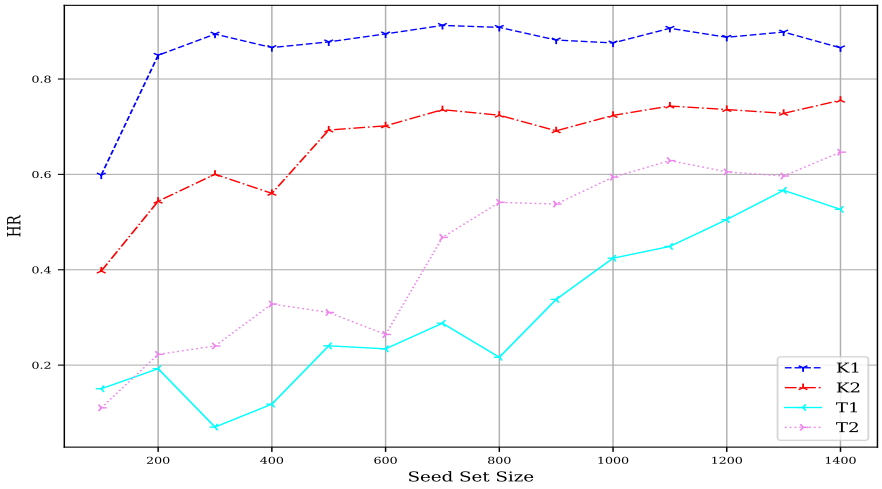

Figure 4.4: HR of the proposed model over four datasets as a function of the seed set size

It should be noted that all of the variants of the proposed model can benefit from the idea of pseudo-relevance feedback [148]. That is, we can retrieve a set of users that are highly relevant to the query and then use them as labeled/seed users to better identify the structure of the target query. This way not only we can leverage the set of unlabeled users but also we can enlarge the set of labeled users. This is in particular more effective if the size of a seed set is relatively small. Of course, the baselines used in the first experiment can also take advantage of the pseudo-relevance feedback. However, the superiority of the IPUD over the baselines does not diminish even if we equip the baselines with the idea of pseudo-relevance feedback. In fact, if we increase the size of the seed set for baselines, their accuracy does not pass a certain level which is significantly lower than the accuracy that is delivered by IPUD. The reason for this is that even the highly relevant users retrieved by baselines may be quite noisy. For example, let's assume a query involves multiple topics and a user is strongly associated with only one of the interest topics. Since the baselines cannot distinguish the topics, the user may be retrieved as a highly relevant user.

4.6.5 Efficiency of IPUD (RQ3)

As mentioned, a key requirement of the user discovery problem is the efficiency of the proposed model. As such, a model with an expensive computational cost like a model that

traverses the entire network does not fit the problem. In this experiment, we evaluated the run time of building IPUD on our datasets as a function of the seed set size. In fact, the purpose of this experiment is to show that IPUD can be used as an online model for user retrieval. The experiments were conducted using python on a 2.3 GHz Dual-Core Intel Core i5 processor with 8GB RAM. As the results show, for relatively large seed sets with 1200 users (and 1200 unlabeled users), the run time over different datasets is less than 25 seconds. Also, the average run time to evaluate a candidate user in our dataset is 5 milliseconds. Hence, even with an average system, thousands of relative users can be retrieved in a few minutes. However, using more powerful computational systems and with the aid of parallelization, a larger scale of users can be retrieved in a matter of seconds.

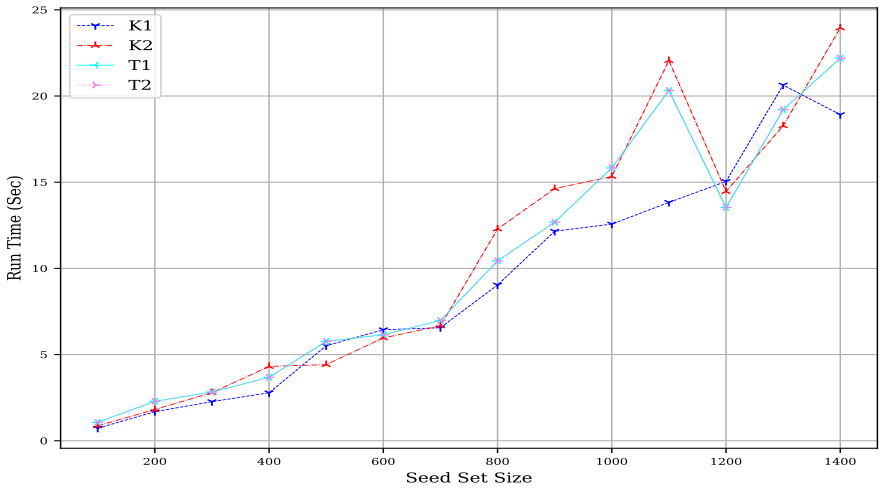

Figure 4.5: Run time of building IPUD model as a function of seed set size over seven datasets

4.6.6 Effectiveness of Candidate Generation (RQ1)

Table 4.3: The percentage of relevant users and irrelevant users with two links towards different ranking intervals of representative nodes

	Rep Nodes rank [0-100]				Rep Nodes rank [300-600]				Rep Nodes rank [600-900]			
	Popularity		Informativeness		Popularity		Informativeness		Popularity		Informativeness	
Topic	**Relevant**	**Irrelevant**	**Relevant**	**Irrelevant**	**Relevant**	**Irrelevant**	**Relevant**	**Irrelevant**	**Relevant**	**Irrelevant**	**Relevant**	**Irrelevant**
Democrats	73.4%	29.1%	44.6%	0.0 %	61.8%	19.3%	42.0%	0.1%	52.7%	14.8%	28.2%	0.1%
NBA	79.6%	22.5%	47.1%	0.0%	44.3%	18.9%	41.8%	0.0%	33.2%	16.2%	30.0%	0.1%
Sports	49.8%	30.6%	25.3%	0.3%	42.6%	17.4%	26.4%	0.1%	37.4%	12.8%	20.1%	0.1%
Technology	54.4 %	32.1%	23.8%	0.1%	46.1 %	21.9%	23.0%	0.1%	40.1%	18.9%	16.3%	0.1%

In this experiment, we evaluated the effectiveness of the candidate generation model. In fact, identifying good candidates is of great importance because it can increase the efficiency of the model.

We collected four sets of 1000 relevant users for each of the interest topics: Democrats, NBA, Sports, and Technology. Moreover, 1000 irrelevant users were randomly selected for each set. We obtained the representative nodes for each dataset, using 1000 relevant users and ranked them according to two methods: popularity and informativeness. Tab. 4.3 shows the percentage of the relevant and irrelevant users that have at least 2 connections towards different ranking intervals of representative nodes: [1-300], [300-600], and [600-900] for each of the sets. For all of the interest sets a larger percentage of relevant users can be retrieved if we rely on more popular representative nodes. On the other hand, the informativeness model retrieves a much smaller percentage of irrelevant users while it also covers a smaller proportion of relevant users compared to the popularity model. In fact, depending on the problem setting, we can choose between these two models. Note that as the results confirm, statistically if rely on less popular neighbors of seed users to collect candidates, a lower proportion of relevant users can be covered. Also, this way of collecting candidate users is more effective than possible alternatives like random sampling or sampling from neighbors of seed users. According to our experiments near to zero percent of our sampled relevant users are the immediate neighbors of the seed users.

4.7 CONCLUSION

We introduced a novel user discovery system to find users with certain interest patterns on Twitter. Indeed, the system can be regarded as a retrieval model to identify users according to an interest-based query, which may have important application in targeted advertising and viral marketing. To the best of our knowledge, this work is the first work to attempt to build such a system.

In order to build the system, we suggested exploiting users follower connection as the data source because users' interests are reflected in their connection and also, unlike content data it is an abundantly available data source for most of the Twitter users. Also, we proposed to define the input queries of the system based on the notion of Query by Example, because of its high descriptive power and its compatibility with the target data source. Based on this setting for the proposed system, we introduced the problem of interest aware seed expansion which is defined as finding users related to a given seed set of users by analyzing the follower/followee connections between the users. Indeed, we cast the task of building the discovery as solving the seed expansion problem.

Our solution to approach the seed expansion problem involves two main steps. First, extracting the linkage patterns shared between seed users and then discovering the other users on the network with similar linkage patterns. We proposed an effective probabilistic

generative model for linkage pattern extraction from seed users' connection based on the notion of representative nodes. We show that the Twitter network is a scale-free network where there is a set of hub nodes associated with each topic of interest. In fact, the hub nodes (representative nodes) associated with an interest topic can be regarded as cluster centers for the cluster of users interested in the topic. The proposed model identifies the structure of a query by finding the cluster centers of the topics involved in the query. Having the cluster centers of the target topics, it retrieves users that belong to the identified clusters. Our experiments were conducted on six datasets collected from Twitter to evaluate the performance of the model. The obtained results confirmed the effectiveness of the model, showing the model's outperformance over the baseline algorithms.

It should be noted the proposed model can be extended from different perspectives. From a data source perspective, to build a more powerful user discovery system, content data can be leveraged along with graph data. We can use content data to augment the input seed set. Obviously, a larger seed set can lead to extract more accurate linkage patterns, which eventually improves the performance of the model. Moreover, we can use content data to better determine the relevance of a user to a seed set. In our model, the relevance prediction is made solely based on graph data. However, we can compare a user's content data to that of the seed user to boost our graph-based relevance prediction model.

From a technical perspective, the proposed ideas in this work can be further generalized. We suggested that the hub nodes (representative nodes) can be regarded as the centers of interest-based clusters in microblogging networks. Hence, the idea of identifying user's interests based on hub nodes can also be considered as a clustering technique. It can be investigated in future works how the structures of interest-based clusters in microblogging networks are different from density-based clusters. Can we use a classic clustering technique to identify interest-based clusters? Additionally, we believe that the idea of query-driven topic modeling can be further investigated for more general settings. The idea can be further developed to build a novel class of topic modeling-based document retrieval algorithms.

CHAPTER 5: PROBABILISTIC SIGN PREDICTION

5.1 OVERVIEW

Various types of interactions between users in diverse domains can be modeled by complex networks, and clearly, not all of such interactions directly indicate users' interests. However, modeling users' interactions, denoted as behavioral profiling, can be used to predict their future interactions, as well as the intention of their interactions [156]. Moreover, based on the notion of homophily, analyzing the interactions of users can aid in finding the proximity/similarity among users, which can eventually be used to build their interest profiles [95]. As mentioned, user models built to predict users' interactions can also take advantage of other data sources associated with users, including content data. While it is beneficial to involve other data sources to build a more accurate interaction prediction model, we solely rely on graph data. This allows us to address the challenges associated with graph data in a more focused way.

In general, the interaction between users may have different types. However, most of the studies about network modeling mainly consider interactions with a single type, i.e., all relations between the nodes have the same concept and meaning. As such, behavioral modeling in such a network is mainly focused on analyzing the existence or absence of the connections between the nodes [157]. However, in OSNs, the nodes may have multiple types of relationships. A class of such relationships can be modeled as networks with positive and negative links where a positive link means friendship (or trust) and negative link corresponds to enmity (or distrust) [16]. In fact, signed networks can be viewed as a representative of networks with multi-type of interactions. We may name Epinions, eBay, Wikipedia and Slashdot as web services from social network domain in which users may have different types of relations (i.e., they are signed networks) [25]. For instance, on eBay, a user can give trust or distrust rates for other active users. On Wikipedia, administrating users give positive or negative votes for the promotion of other users.

In networks with multi-type of relations, not only the existence of the link between two nodes is of great importance, but also its label is as well. In fact, predicting the labels of links becomes a crucial problem for analyzing the future interactions of users [158]. The problem of inferring labels of relations and in particular sign of a relationship has attracted attention in recent years [24]. As mentioned, sign prediction has various applications in various domains. For example, user-to-user recommender systems can benefit from efficient sign prediction algorithms [159]. Also, such algorithms can also be used for intention modeling, e.g., to

identify malicious users in signed networks [160].

Clearly, to solve the sign prediction problem, we seek to construct a model that is efficient as well as effective in terms of accuracy. Moreover, it is desirable to have a model that can incrementally adapt to changes in network structures because a growing number of emerging business, where high rate streams of detailed data are constantly generated, necessitate the need to build such models. Finally, it would be an interesting capability of a model to be able to perform on networks with more than two types of relations.

For sign prediction, we are facing the major challenge of the *sparisty* of information that plagues any complex real-world network [161]. Indeed, this is a key issue when deciding what information sources should be exploited for addressing the prediction task. In the sense of granularity, structures within a network can be viewed on two levels; local structures and global structures [11]. Roughly speaking, local structures are often described as structures at the level of the paths surrounding target edges while global structures refer to connectional patterns between communities of nodes.

One the one hand, some existing works solely rely on local information to predict the sign of the edge in question which makes them vulnerable to the sparsity problem. A representative example of such models is a supervised learning-based predictor proposed by Leskovec et al. [25]. The model, first extracts a set of features from the triangles involving the target edge; then, a learning method is employed to build a classifier to solve the sign prediction problem [25]. The proposed features in this work were derived based on two theories in signed networks, known as balance theory and status theory [162]. Both balance and status theories specify the patterns of connection at the level of triangles. However, robustness of the model against sparsity problem is quite limited as many nodes in networks do not share common neighbors [25]. To remedy the problem, Chiang et al. defined a new set of features by generalizing the notion of balance theory to l-cycles where a l-cycle is a path with length l from a node to itself. In fact, by increasing l, the extracted features become less localized [163]. Although the model addresses the problem to some extent, still it is not able to capture global structure at the level of communities; moreover, the feature extraction process is computationally expensive, and the number of features exponentially increases with the length of cycles [11]. Very recently, a new line of research has aimed to solve the problem with the aid of embedding models [33, 164]. However, these models are not equipped to leverage the valuable information hidden in global structures because their proximity measures are not able to preserve global structures [43].

On the other hand, other existing works seek to use global structure to approach the problem which brings them robustness toward sparsity. It has been shown that if local structures in a signed network follow balance theory, it leads to a clusterability property of signed

network stating that a complete network is weekly balanced if it can be partitioned into K clusters in a way that all the edges between clusters are negative and all the edges within clusters are positive [26, 161]. With the help of this property, Hsieh et al. showed that signed networks exhibit a low-rank structure and formulated the sign prediction problem as a low-rank matrix completion problem [158, 161]. Having this formulation, available algorithms for matrix completion can be used for sign prediction. In another work, again based on clusterability property derived from balance theory, a prediction model was developed by finding the similarities of connectional patterns between clusters [26]. Naturally, cluster-based models are more robust against sparsity, and the obtained results from these works confirm that global structures may indeed have substantial merits for prediction purpose [26, 161]. However, such models are not able to extract information form local structures which can negatively affect their prediction accuracy.

In all, none of the previously introduced models fully leverage the rich information contained in both local and global structures. Besides this major issue, most of them have been designed specifically for signed networks as they have a basis in theories that are applicable only on signed network and extending such models for the general problem of link label prediction is non-trivial [25, 26, 163]. The other issue with many of the previous methods is that they cannot automatically adapt their prediction models to real-time data streams. In other words, to update those models based on the updated structure of input networks, prediction models should be built from scratch.

Towards addressing the prediction task effectively as well as solving the sparsity problem, as our key insight, we propose to holistically integrate and exploit both local and global information on a network. We suggest that indeed there exists a dilemma between these two information sources. Intuitively, local information specifically describes what is happening between target nodes, but the information obtained from local structures may end up being unreliable due to data sparseness. One the other hand, global structures tend to provide reliable information; however, naturally, such structures could not capture detailed information about target links. Indeed, it is challenging how to solve this dilemma between local and global structure optimally, i.e., when to use local structures and when to rely on global structures. A model that can solve this dilemma and exploit both information sources successfully not only would be able to address the sparsity issue but also can achieve promising results in terms of accuracy.

In this approach, we propose the concept of statistical link label model, which can be defined as a probability distribution over the possible labels of the target edge, given the connections pointing out from the initiator node. The key characteristic of this model which enables us to exploit both local and global structures is that it reserves some parameters

for the use of sophisticated smoothing techniques. For estimating those parameters, we develop a cluster-based smoothing model through which we bridge the gap between local and global structures. This model considers global structure as a background knowledge and uses the background knowledge when sufficient local information is not available. Indeed, adaptivity of the proposed models comes from Dirichlet smoothing which enables us to make the balance between local and global structures. To build up the background knowledge, a novel clustering algorithm is developed. The proposed objective function used in this clustering algorithm is consistent with link label prediction task and can be optimized with the aid of optimization algorithms such as Gibbs sampling [165].

We show that the proposed model not only leverages both local and global structures but also addresses the aforementioned weaknesses of the counterpart models. We evaluate the models on three benchmark signed network datasets. Our results reveal that the proposed models consistently outperform previous sign prediction algorithms and demonstrate the high tolerance of the model toward the sparsity problem. We conjecture the main reason that accounts for the results is that the proposed link label model provides a principled way of exploring the local-global structures, which also makes the proposed predictors capable of adjusting themselves to the sparsity level of input networks. Moreover, the proposed model have lower computational complexity than the counterpart models. Also, the models can be updated incrementally with linear computational complexity on the total number of edges added to the input network. Indeed, the proposed model performs based on extracting some simple statistics from the network which accounts for its high efficiency. In addition, the model can be easily extended to perform on networks with more than two types of relations. Lastly, the structure of the proposed models have this advantage over the comparable models that it can easily accommodate contextual information and node attributes to provide more accurate predictions.

5.2 PRELIMINARIES

5.2.1 Problem Formulation

In this section, we provide a formal definition of the signed prediction problem. A signed network is defined as a directed graph $G(V, E)$ with a link type mapping function $\varphi : E \to A$, in which all nodes $v \in V$ are of the same type, each link $e \in E$ belongs to one particular link type $\varphi(e) \in A$ where $A = \{+, -\}$. Suppose that over this network labels for some of the edges are missing. The problem is formulated as predicting the missing label for a given edge $e(u, v)$ from u denoted as *initiator node* to v denoted as *receiver node*, using the information

on the labels from other edges. Note that link label prediction problem is the general case of the sign prediction problem where link labels may have more than two classes.

5.2.2 Dilemma Between Local and Global Structures

Data sparsity is a prominent and critical issue when designing prediction models over networks. The performance of such models relies upon accurate and sufficient information being found in a network. However, it is known that abundantly sufficient data is not always available, in particular, on networks with high levels of sparsity. Imagine we are given a signed network, and the task is to predict the sign of the edge between two nodes contributing in quite a few links and sharing no common neighbors. Apparently, if we only rely on local information to determine the sign of the edge, we might make unreliable and inaccurate predictions due to lack of local information.

In general, it has been shown that for a large proportion of nodes in networks it is practically not possible to make reliable and accurate predictions by only relying on local structures. For example, in previously proposed path-based models, it has been shown that prediction accuracy is considerably lower for the edges with lower values of embeddedness [25]. Embeddedness somewhat indicates the amount of local information available for prediction. We may draw this conclusion that, for the models based on local information, prediction cannot be performed accurately if local information about the target edge is not sufficient.

As a typical solution to the sparsity problem, the global structure of the network can be taken into account and patterns of connections between clusters may be utilized to approach the problem. Even on a network with high levels of sparsity, it is possible to assign a user to the cluster of users with similar connectional patterns and build a model based on the patterns of links between clusters. Intuitively, pattern extraction between clusters is less likely to encounter the sparsity problem because there always exist a substantial number of edges among clusters.

However, despite the fact that cluster-based models solve the sparsity problem, but they perform based on patterns that less specifically reflect the individual connections between target nodes. To our knowledge, most of the previously introduced models have chosen either local or global structures to build up their models. Indeed, models based on global structures may lose some rich and valuable local information while the predictors based on local structures might suffer from the sparsity problem.

This weakness of the previous models in fully capturing the local and global structures indicates a dilemma between reliability provided by global structure and specificity delivered

by local data. A model that solves this dilemma in an intelligent manner not only would be able to tolerate against sparsity but also might outperform the counterpart models because it attempts to leverage more data.

5.2.3 Solving the Dilemma

In this thesis, we propose a model that resolves the dilemma between global and local information; a model that can combine the information obtained from local and global structures in a principled way. This model can be viewed as a statistical model that enables us to compute the distribution of possible labels for the target edge given its *context* where context can be defined as the edges that their labels are somewhat related to that of the target edge. By computing such a distribution, it would be straightforward to infer the label of the edge in question.

The structure of the proposed model reserves some parameters for the use of sophisticated smoothing models which allows us to combine local and global structures in a principled way. By taking advantage of this characteristic, we propose the concept of cluster-based smoothing on networks. The main idea behind this concept is that the patterns extracted from the global structure of a network can be assumed as *background knowledge* on individual nodes. For estimating the parameters, we can more rely on this background knowledge as less reliable information can be gained from local structures. This concept of smoothing provides a rigorous theoretical foundation for integrating local and global analysis of networks and allows us to build prediction models that are tolerable and adaptive to sparsity.

It is worth mentioning that the structure of the proposed model and the cluster-based smoothing technique used in the model have some conceptual connections to statistical language models [166]. A language model can be described as a function to estimate the probability distribution for an upcoming word w given its context, $P(w|context)$, where context is often defined as a sequence of previous or next words. Similar to our proposed models, one important characteristic of statistical language models such as n-gram model is their tolerance to data sparseness achieved by being able to employ smoothing techniques. For example, it has been shown that clustral structures of documents can be easily injected into such models through smoothing techniques [167].

5.3 STATISTICAL LINK LABEL PREDICTION

The sign prediction problem can be approached by developing a probabilistic model to

estimate the probability distribution over each of the classes of the target edge given the input graph. By comparing the obtained probabilities, the class with the highest probability may be defined as the final prediction:

$$\hat{l} = \arg\max_{l} P(\varphi(e(u_i, u_j)) = l | G(V, E)) \tag{5.1}$$

In this thesis, we follow this direction and aim to approach the prediction model by proposing a probabilistic model named as *link label model* that estimates the probability distribution over possible labels for a link given its *context*, $P(\varphi(e(u_i, u_j)) = l | context)$.

If we view a link label model as a probabilistic classifier, the concept of context may be treated as the features of the classifier that should be extracted from the input graph $G(V, E)$. Now, the question is how to specify the context of the target edge. We suggest that the context of $e(u_i, u_j)$ can be defined as the set of the labels of connections initiated by u_i. Formally, context of $e(u_i, u_j)$ is denoted as $\varphi(E_i)$ where $E_i = \{e(u_i, u_x) | e(u_i, u_x) \in E\}$.

In language models, context of a word is typically defined as the window of words with a certain size before or after the word. We believe that analogous to the sequence of the words in a sentence, there exists a dependency between the labels of connections made by a user.

Now having the definition of context, the task is reduced to estimating the target distribution given the defined context. One very trivial approach to build the model is to generate all the possible contexts and build a model that has a parameter for every possible pair of the generated contexts and the target links. Clearly, with this approach, the number of parameters would be quite unmanageable.

This leads us to make some assumptions to build a simpler model with less parameters. As the simplest model, we may assume the labels of connections pointing out from a node are generated independently. Although such a model is simple, it makes an unrealistic assumption on the occurrence of labels of links. This assumption implies that when a user u_i establishes a link toward u_j, the sign of the edge only depends on node u_j, i.e., in this model we have $P(\varphi(e(u_i, u_j)) = l | \varphi(E_i)) = P(\varphi(e(u_i, u_j)) = l)$. Therefore, we need to design a more complex model that is able to take the context of the target link into account.

Note that while, theoretically speaking, we would prefer to build a sophisticated model that can model occurrences of links more accurately, in reality, we face a tradeoff; as the complexity of model increases, the number of parameters increases as well. Therefore, the simplicity of the model is a factor that plays a crucial role in the final performance of the model.

In the next subsections, two link label models based on local structures are presented.

Then, we discuss how these two models can be modified to capture global structures. Finally, we introduce the concept of smoothing which allows us to extend the proposed models so that they can leverage both local and global structures.

5.3.1 Link Label Modeling Based on Local Structures

We propose two link label models based on local structures. The first model named as local structure-based target link generator model (LTLGM) directly estimates the target distribution from the given context. However, the second model called as Local structure-based Context Generator Model (LCGM) attempts to solve the reverse problem. That is, we can estimate the probability of generating the context given each possible label for the target edge and then assign a label to the target edge that is more likely to generate the context.

Target link label generator model. As stated, LTGM directly estimates the target distribution from the given context. This model can also be viewed as a probabilistic discriminative classifier. Discriminative classifiers model the posterior $p(y|x)$ directly, or learn a direct map from inputs x to class labels y. However, it is not feasible to build a model through enumerating all the possible combinations of link labels generated by a user due to that such a model has a potentially infinite number of parameters. In order to tackle this issue, instead of capturing the dependency between the entire context and the target link, we can capture the dependencies between each element of the context and the target link. As such, the target probability can be calculated as follows:

$$P(\varphi(e(u_i, u_j)) = l|\varphi(E_i)) = \sum_{u_x \in H_{u_i}} \pi_x P(\varphi(e(u_i, u_j)) = l|e(u_i, u_x))), \tag{5.2}$$

where H_{u_i} is the set of users that u_i is pointing out to, i.e., the heads of the links initiated by u_i, and π_x denotes the weight of the model associated with u_x. In our experiments, we set $\pi_x = \frac{1}{|E_i|}$. Also, $P(\varphi(e(u_i, u_j)) = l|e(u_i, u_x))))$ can be estimated using Maximum Likelihood Estimation (MLE) and based on local structures:

$$P_{local}(\varphi(e(u_i, u_j)) = l|e(u_i, u_x))) = \frac{|T_{u_j,l} \bigcap T_{u_x,\varphi(e(u_i,u_j))}|}{|T_{u_j} \bigcap T_{u_x,\varphi(e(u_i,u_j))}|}. \tag{5.3}$$

where T_{u_x} is the set of nodes that have a connection toward u_x, i.e., the tails of the edges pointing to u_x, and $T_{u_x,\varphi(e(u_i,u_x))}$ is a subset of T_{u_x} in which the labels of the edges are the

same as $\varphi(e(u_i, u_x))$. In fact, in this equation we calculate the proportion of nodes that have links to both u_x with label $\varphi(e(u_i, u_x))$ and u_j with label l to those that have links to both u_x with label $\varphi(e(u_i, u_x))$ and u_j with either positive or negative labels. Ideally, it is desirable to estimate dependency probabilities in respect to the target initiator node. That is, we can first identify the users similar to the initiator node u_i, and then estimate the dependency probability by only considering the connection of such users. However, this strategy may reduce the number of available statistics to estimate the target probability.

It is worth mentioning that the structure of the proposed model is conceptually related to the concept behind k-skip-n-gram models where a certain distance k allows a total of k or less skips to build the n-gram model [168].

Context generator model. Since our ultimate goal of estimating target distributions is to address the prediction problem, instead of solving the original problem, the reverse problem may be solved. That is, we can estimate the probability of generating the context given each possible label for the target edge and then assign a label to the target edge that is more likely to generate the context. Following this idea, we propose another link label model named as Local structures-based Context Generator Model (LCGM).

From another point of view, we can think of the context generator model as a generative classifier. In general, generative classifiers learn a model of the joint probability of inputs x and labels y denoted as $p(x, y)$, and do predictions by employing Bayes rules to compute $p(y|x)$ and finally choosing the most likely label y.

Using Bayes theorem, the probability of the label l for the target edge $e(u_i, u_j)$ conditioned to $\varphi(E_i)$ can be computed as follows:

$$P(\varphi(e(u_i, u_j)) = l|\varphi(E_i)) = \frac{P(\varphi(E_i)|\varphi(e(u_i, u_j)) = l)P(\varphi(e(u_i, u_j)) = l)}{P(\varphi(E_i))}, \tag{5.4}$$

where $P(\varphi(E_i)|\varphi(e(u_i, u_j)) = l)$ is the probability of observing $\varphi(E_i)$ conditioned to the fact that $\varphi(e(u_i, u_j)) = l$ and $P(\varphi(E_i))$ represents the probability of observing $\varphi(E_i)$. Since this is a classification task and $P(\varphi(E_i))$ is a constant for all of the classes, it can be removed from the prediction process. $P(\varphi(e(u_i, u_j)) = l)$ is our prior knowledge about class l which can be set to a uniform distribution. Setting $\varphi(e(u_i, u_j)) = l$ to the uniform distribution, Equation 5.4 can be reduced to:

$$P(\varphi(e(u_i, u_j)) = l|\varphi(E_i)) \simeq P(\varphi(E_i)|\varphi(e(u_i, u_j)) = l). \tag{5.5}$$

Now the task is to compute $P(\varphi(E_i)|\varphi(e(u_i,u_j))=l)$. Calculating this probability is not trivial, because the space of possible connections for each node is too vast. Therefore, we assume that the labels of the links within E_i are independent given $\varphi(e(u_i,u_j))=l$. This simplification reduces Equation 5.5 to:

$$P(\varphi(e(u_i,u_j))=l|\varphi(E_i)) \simeq \prod_{u_x \in H_{u_i}} P(\varphi(e(u_i,u_x))|\varphi(e(u_i,u_j))=l), \tag{5.6}$$

where $P(\varphi(e(u_i,u_x))|\varphi(e(u_i,u_j))=l)$ is the parameter of the model which can be estimated using MLE and by exploiting local structures as follows:

$$P_{local}(\varphi(e(u_i,u_x))|\varphi(e(u_i,u_j))=l) = \frac{|T_{u_j,l} \bigcap T_{u_x,\varphi(e(u_i,u_j))}|}{|T_{u_x} \bigcap T_{u_j,l}|}, \tag{5.7}$$

5.3.2 Link Label Modeling Based on Global Structures

In the previous section, we introduced two models that only take local structures into account to estimate their parameters. However, as stated, parameter estimation based on local structures is quite vulnerable to sparsity problem. Indeed, despite the fact that we provided some robustness in our prediction models against sparsity by imposing a few assumptions when designing the structure of our models, but we find that still, we encounter sparsity issue when estimating the parameters. Obviously, the estimations made in Equations 5.3 and 5.7 are reliable if the number of available samples is sufficient, otherwise, the unreliability of estimation would lead to inaccurate predictions. However, there is no guarantee for the availableness of such sufficient number of samples. This motivates us to develop link-label models that exploit global structures.

Following this idea, we adapt the proposed models so that they can perform based on global structures.To this end, we incorporated the notion of clustering. By leveraging graph clustering algorithms, one can put users/nodes with similar connectional patterns in the same clusters whereas nodes with different patterns of connection are in different clusters. We suggest that the model's parameters can be estimated by analyzing connection at the level of clusters rather than individual nodes. The intuition behind this idea is that in the absence of sufficient data between nodes we can focus on the link between the clusters they belong to because nodes within the same cluster are expected to exhibit similar connectional patterns. Clearly, the way we partition the network into clusters plays a key role in our model. In the next section, we will present the proposed clustering algorithm which has been specifically designed for this task. Based on the proposed idea, given a set of clusters

over the network the parameters of the model specified in Equation 5.3 can be estimated as follows:

$$P_{global}(\varphi(e(u_i,u_j))=l|\varphi(e(u_i,u_x)))=\sum_{u_x\in E_i}\pi_x\frac{|T^{\Omega(u_i)}_{(\Omega(u_x),\varphi(e(u_i,u_x))}\bigcap T^{\Omega(u_i)}_{(\Omega(u_j),l)}|}{|T^{\Omega(u_i)}_{(\Omega(u_x),\varphi(e(u_i,u_x))}\bigcap T^{\Omega(u_i)}_{(\Omega(u_j))}|}, \tag{5.8}$$

where Ω is a cluster mapping function $\Omega : V \to C$ in which each node $v \in V$ belongs to one particular cluster $\Omega(v) \in C$ and $T^{\Omega(u_i)}_{\Omega(u_x),\varphi(e(u_i,u_x))}$ represents nodes belonging to $\Omega(u_i)$ that have links toward the nodes within $\Omega(u_x)$ with label $\varphi(e(u_i,u_x))$.

Note that, in order to estimate the probabilities based on local structures, the entire set of users in the graph are taken into account due to the sparsity problem. However, to estimate the parameters based on clusters we do not face sparsity, and hence the probability may be estimated with respect to the initiator node. As it can be seen in Equations 5.8, only nodes that belong to the same cluster as the initiator node are considered to estimate the model.

In the rest of this section, we refer to this model as Global structure based Target Link label Generator Model (GTLGM).

In the same fashion, to build a context generator model that benefits from the global structures, the parameters specified in equation 5.7 can be estimated as follows:

$$P_{global}(\varphi(e(u_i,u_x))|\varphi(e(u_i,u_j))=l))=\frac{|T^{\Omega(u_i)}_{\Omega(u_x),\varphi(e(u_i,u_x))}\bigcap T^{\Omega(u_i)}_{\Omega(u_j),l}|}{|T^{\Omega(u_i)}_{\Omega(u_x)}\bigcap T^{\Omega(u_i)}_{\Omega(u_j),l}|}, \tag{5.9}$$

We name the method constructed based on this estimator as Global structure-based Context Generator Model (GCGM).

5.3.3 Integrating Local and Global Structures

In previous sections, we introduced models that either rely on global or local structures. However, as stated, our main goal in this section is to propose a model able to solve the dilemma between these two information sources. To achieve this goal, we employ the concept of smoothing on graphs which can effectively solve the problem. We believe that the notion of smoothing can be systematically applied to graph mining algorithms and may be viewed as a critical component of probabilistic graph mining models.

The major principle in smoothing methods is that a sparsely estimated conditional model can be smoothed using a more densely estimated but simpler model. We consider the estimators based on local structures as sparsely estimated models, and treat the models based

on clusters as simpler models with more reliable estimations. Therefore, estimations based on individual nodes can be smoothed with the estimations from global structures to build models that benefit from both of the information sources. Various methods have been introduced for smoothing such as Dirichlet and Katz for different purposes over different domains [169]. Here we specifically focus on Dirichlet model. The reason for this is that Dirichlet smoothing is capable of regulating the estimation obtained from the estimators according to the reliability provided by the sparsely estimated model. By employing Dirichlete smoothing, the estimator of LTLGM can be smoothed through the one introduced for GTLGM. As such, we define a smoothing based estimator as follows:

$$\begin{aligned} P(\varphi(e(u_i,u_j)) = l|\varphi(e(u_i,u_x))) = \\ (1-\lambda)P_{local}(\varphi(\varphi(e(u_i,u_j)) = l)|e(u_i,u_x))+ \\ \lambda P_{global}(\varphi(e(u_i,u_j)) = l|\varphi(e(u_i,u_x))), \end{aligned} \tag{5.10}$$

where P_{local} and P_{global} are presented in Equations 5.3 and 5.8 respectively, and λ is defined as:

$$\lambda = 1 - \frac{|T_{u_x} \bigcap T_{u_j,l}|}{|T_{u_x} \bigcap T_{u_j,l}| + \mu}. \tag{5.11}$$

where μ is a parameter for Dirichlete smoothing that should be tuned. In fact, λ indicates the reliability of the estimations made based on local structures. We name the model obtained based on this estimator as Smoothing-based Target Label Generator Model (STLGM).

The same idea can be employed to build a context generator model by integrating local and global structures. Formally, we define a smoothing-based estimator:

$$\begin{aligned} P(\varphi(e(u_i,u_x))|\varphi(e(u_i,u_j)) = l) = \\ (1-\lambda)P_{local}(\varphi(e(u_i,u_x))|\varphi(e(u_i,u_j)) = l)+ \\ \lambda P_{global}(\varphi(e(u_i,u_x))|\varphi(e(u_i,u_j)) = l), \end{aligned} \tag{5.12}$$

in which P_{local} and P_{global} are specified in Equations 5.7 and 5.9 respectively, and λ' is defined as:

$$\lambda' = 1 - \frac{|T_{u_j} \bigcap T_{u_x,\varphi(e(u_i,u_x))}|}{|T_{u_j} \bigcap T_{u_x,\varphi(e(u_i,u_x))}| + \mu}. \tag{5.13}$$

The model from this estimator is referred as Smoothing-based Context Generator model.

5.4 CLUSTERING

The way we do clustering is a crucial factor that decides the performance of the smoothing

algorithm. For example, one may randomly partition the network and then derive the probabilities from those randomly generated clusters. By adding the prior model obtained from such method of clustering, we would add some noise to our prediction model that may lead to inaccurate predictions.

As stated, in our models based on global structure, we suggested that to extract connectional patterns between individual nodes, in the absence of enough local data, the connections between the clusters they belong to can be analyzed. This implies that nodes within the same cluster share the similar connectional patterns and can be considered as structurally *equivalent* units. Therefore, to build our clustering model first, we specify the notion of equivalence between two nodes.

Here, we suggest that two nodes are *equivalent* if they have identical types of connections to equivalent others. For example, imagine that u_a is equivalent to u_b and u_c is equivalent to u_d, and there exist two edges $e(u_a, u_c)$ and $e(u_b, u_d)$. The proposed notion of equivalence imposes $\varphi(e(u_a, u_c))$ must be the same as $\varphi(e(u_b, u_d))$.

With this definition, if we partition the network into clusters of equivalent nodes, all of the edges between every pair of the clusters would have identical labels. Indeed, here clusters can be viewed as *roles* carried by nodes. Imagine there are K latent roles in a network and every node within the network is assigned to one of them and the relation between two nodes is decided by the roles they hold. It means that for example in a signed network, all of the edges from the nodes with role A to nodes with role B must have the same label. Having such a structure over the input network, we can infer a node's connectional pattern by analyzing the link patterns of the node with the same role as the target node.

For partitioning the network based on our definition of equivalence, we should solve an optimization problem where the task is to assign nodes to clusters so that the edges from one cluster to another have identical labels.

Nevertheless, it must be mentioned that practically it is not possible to achieve such an ideal set of clusters over the input network which means that we need to relax our goal of clustering. Therefore, the **objective of clustering** is defined as partitioning the network, so that the majority of the links from one cluster to another have the same label. That is, suppose that an edge's label from one cluster to another is a random variable with a probability distribution over its possible values: positive and negative which means that for a network with K clusters we would have K^2 random variables, one random variable for each pair of the clusters. We seek to partition the network in a way that the entropy of these random variables is minimum. We define our objective function as the weighted summation of the entropies of the random variables and the weight of each random variable is determined by the number of the edges between the clusters associated with the random

variable. Formally, the objective function for a cluster set g over a signed graph G is defined as follows:

$$\phi(g) = -\sum_{c_i \in g} \sum_{c_j \in g} |E_{c_i,c_j}| \sum_{k \in \{+,-\}} p_{c_i,c_j}(k) log(p_{c_i,c_j}(k)) \tag{5.14}$$

where $|E_{c_i,c_j}|$ is the size of the set of the edges from cluster c_i to c_j and $p_{c_i,c_j}(k)$ indicates the probability that an edge from cluster c_i to cluster c_j has label k. Using ML estimator, this probability can be computed as:

$$p_{c_i,c_j}(k) = \frac{|E_{c_i,c_j}(k)|}{|E_{c_i,c_j}|} \tag{5.15}$$

where $|E_{c_i,c_j}(k)|$ is the size of the set of the edges from cluster c_i to c_j with label k.

In order to do clustering based on the introduced objective function, a potentially difficult optimization problem must be solved. Indeed, our task is a special case of the general problem of finding the minimum of an objective function over a large combinatorial set. This type of problems is often amenable to Markov Chain Monte Carlo (MCMC) optimization, in which the model runs a Markov chain on the target combinatorial set and evaluates objective function at the successive states [165]. We employed an optimization model to construct our clustering algorithm. In the next subsection, we describe the optimization model.

5.4.1 Clustering Based on Gibbs Sampling

Our clustering algorithms can be described in two steps. First, we need to map the input combinatorial set to a graph that represents the neighborhood structure of the combinatorial set. Secondly, by making a random walk over the graph, one can reach a partition that fits the objective function under consideration [165]. In the following, we first define the neighborhood structure and then describe how a random walk can be made over this graph using Gibbs sampling algorithm.

Having our objective function, $\phi : P \rightarrow R+$, where $P = \{g_1, g_2, ..., g_m\}$ is the set of the all possible partitions on input graph and m is the number of all possible partitions on the graph, we can define the neighborhood structure of P as follows. Let's consider GP as a connected, undirected graph with vertex set P in which there is an edge between two vertices, g_i and g_j, if and only if it is possible to get from partition g_i to partition g_j by moving exactly one of the n objects in g_i to a different cluster. In fact, the graph GP represents a neighborhood

structure on P; in which g_i and g_j are neighbors if and only if they have an edge in common. We denote the number of neighbors of the vertex g in GP as $d(g)$. Now the question is how to manage the random walk on graph GP.

As states, we employ a Gibbs sampler to make a random walk over GP. Suppose the current state of the Markov chain is the partition g_i, and v_j a fixed object within the input graph. Let's suppose we remove the object v_j from the partition g_i, and represent the obtained partition as $g_{i,-j}$ and define $S_{i,-j}$ as the set of partitions in P that, when the object v_j is removed are identical to $g_{i,-j}$. In our Gibbs sampling approach, to select a new state, we first evaluate the objective function for all partitions within $S_{i,-j}$ and then draw the new state from $S_{i,-j}$ based on the probabilities proportional to the objective function. In each iteration, the target object v_j should be varied in order to cover all the objects within the network. There are two strategies to select the target object at each iteration: deterministic Gibbs sampler and random scan Gibbs sampler. In the deterministic version, v_j is selected in a predetermined order while in the random scan version v_j is selected randomly from N.

5.5 COMPUTATIONAL COMPLEXITY

In this section, we briefly discuss the computational complexity of the proposed prediction model. As stated in previous sections, in order to make predictions using our local structured based models, we first need to find the list of outgoing edges from target initiator node and then estimate the probabilities introduced in equations 5.3 and 5.7. Note that these parameters are estimated based on the number of nodes that have connections to both u_x with label l and u_j with label l'. Those statistics of the input network can be computed in an offline manner and we don't need to wait to calculate them until the prediction time. Therefore, they can be stored in a matrix named as Node-based Aggregation Matrix (NAM) with the size $|V|^2 \times 2 \times 2$ in which element $a_{m,n,l,l'}$ is the number of nodes that have a connection toward u_m with label l and a connection toward u_n with label l'. NAM can be extracted from the network by one pass of scanning the edges and hence the computational complexity of building NAM is $O(E)$. Also, if we add E' edges to the network, the cost of updating NAM would be $O(E')$.

To add the smoothing part to the basic prediction models, we should first partition the network and then estimate the parameters based on global structures as stated in equations 5.8 and 5.9. The computational complexity of clustering algorithm based on Gibbs sampling algorithm is $O(EI)$. According to our experiments, practically, the optimization algorithm converge after $10 \sim 20$ iterations on our benchmark networks.

To estimate the parameters based on global structures, again we can use statistics that

could be calculated in an offline manner. Those statistics can be obtained from a Matrix named as Cluster-based Aggregation Matrix (CAM) with the size $|C|^3 \times 2 \times 2$, in which $a'_{s,m,n,l,l'}$ is the number of nodes from cluster C_s that have connections to cluster C_m with label l and connections toward cluster C_n with label l'. CAM can be constructed by one pass through the edges within the network. Overall, the computational complexity of constructing NAG and CAG is $O(E)$ and the complexity of partitioning the network is $O(EI)$. As mentioned one advantage of the proposed models is that they can be updated incrementally based on a live stream of data with linear complexity on the number of edges added to the underlying network. Imagine a prediction model is constructed based on an input network and we aim to update the model incrementally using a live stream of data containing new edges added to the networks. To do so, the NAM and CAM graphs can be updated by adding up to the weights of the links corresponding to the newly added edges. Obviously, this could be done by one pass through the new edges. In other words, to update the model we do not need to build the model from scratch. However, this is not the case for most of the counterpart models. Also, we claimed that the proposed model is a generic model for link label prediction. In this thesis, the model is introduced on the basis of the sign prediction problem. However, the model can be easily extended to perform on networks with multi-type of relations. Consider the equation 5.3 in which the label for an underlying edge is predicted by finding the probability of positive and negative labels for the edge given its context. To do label prediction for a network with $|K|$ possible labels, we simply need to calculate the probability of each of the $|K|$ possible labels given the context and finally select the label with the highest probability. Clearly, the clustering algorithm should also be modified accordingly for networks with multi-types of relations. Note that, the comparable sign prediction algorithms do not possess this flexibility.

5.6 EXPERIMENTS

In this section, we compare the performance of the proposed link label prediction model with those of the state of the art. The performance of the methods is assessed over three real-world datasets: Epinions, WikiElection, and Slashdot. We also investigate the validity of the idea of leveraging global structure to remedy the sparsity problem through our experiments.

5.6.1 Test Methodology

We evaluate the performance of the proposed algorithms using 10-fold cross validation as the testing algorithm. In 10-fold cross validation, in each fold, 10% of the original dataset

is considered as test dataset and 90% as training dataset. Accuracy of classification is a common metrics to evaluate the predictions. However, in our datasets, the number of positive edges is considerably greater than the number of negative edges, and comparing prediction methods based on the accuracy of original test sets could be misleading under some circumstances. Thus, balanced accuracy is used in our experiments as the evaluation metric. In this method, the experiments are performed on the original test and training datasets and mean of the true positive rate of the predictions on two classes is reported (+1 and -1 for positive and negative signs, respectively).

Table 5.1: Datasets statistics

	Nodes	Edges	+Edges	-Edges
Epinions	119217	841200	85.0%	15.0%
Slashdot	82144	549202	77.4%	22.6%
WikiElection	7118	103747	78.7%	21.2%

5.6.2 Datasets

In order to assess the performance of proposed methods, we tested them on three signed networks: Slashdot, Epinions, and WikElection which have frequently been used as benchmarks in different works [24]. Table 1 provides some statistics on these datasets.

- **Epinions** is a web service about online product review in which users can express their votes on products. Also, in this platform users can give positive (trustworthy) or negative (untrustworthy) votes to other users regarding their reviews on products. The relations among users can be modeled by a directed signed network where users are nodes and their votes on each other form directed signed links of the network.

- **Slashdot** dataset is obtained from a web service (technology news website) in which users can flag each other as friend or foe in order to indicate their approval or disapproval of their comments.

- **WikiElection** dataset is the network obtained from users' votes for elections of administrators in Wikipedia. In Wikipedia election, users may give positive or negative votes for the promotion of other users. Evidently, the meaning of links and their sign in WikiElection dataset differs from the other two datasets. However, all three systems can be modeled and generalized by signed networks, and the obtained network can be used to analyze the original systems.

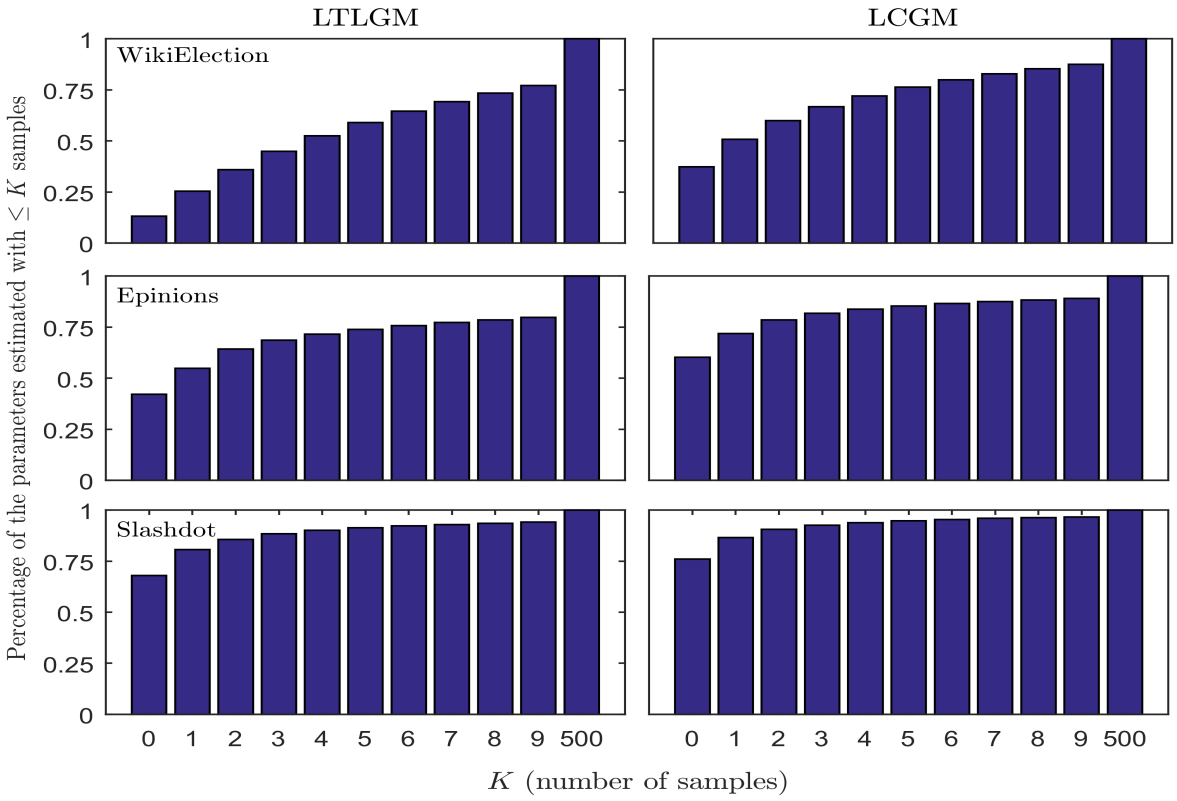

Figure 5.1: Percentage of the parameters of LTLGM and LCGM estimated with less than K samples as a function of K over WikiElection, Epinions and Slashdot datasets.

5.6.3 Reliability of Parameter Estimation for LTLGM and LCGM

To predict the label of the link from an initiator node u_i with $|E_i|$ outgoing links toward a receiver node u_j using LTLGM and LCGM $|E_i|$ and $2|E_i|$ parameters should be estimated, respectively. As mentioned, the reliability of estimating these parameters is a crucial factor in the accuracy of the predictions made by these models which can be defined by the number of available samples for estimation. The more samples available for estimation the higher the reliability of the estimated parameters.

In this experiment, we investigate whether a sufficient number of samples are available for estimating the parameters of LTLGM and LCGM over our benchmark networks. We applied the basic models on the test sets obtained from WikiElection, Slashdot and Epinions datasets and measured the number of available samples for estimating the target parameters. Fig. 5.1 demonstrates the percentage of the parameters estimated with less than K samples over the three datasets as a function of K. As it can be seen from the figure, a large proportion of parameters for all three datasets should be estimated based on a few samples.

For example, more than 40%, 60% and 80% of the parameters on WikiElection, Epinions and Slashdot datasets for LTLGM should be estimated with less than four samples. On Slashdot dataset, for more than 60% of parameters, even a single sample does not exist to estimate the parameters.

The problem is more severe for LCGM, as in LCGM we need to estimate two times more parameters than LTLGM. These results clearly show that how sparsity problem can negatively affect our local structure-based prediction models and it necessitate a principled model to deal with sparsity problem.

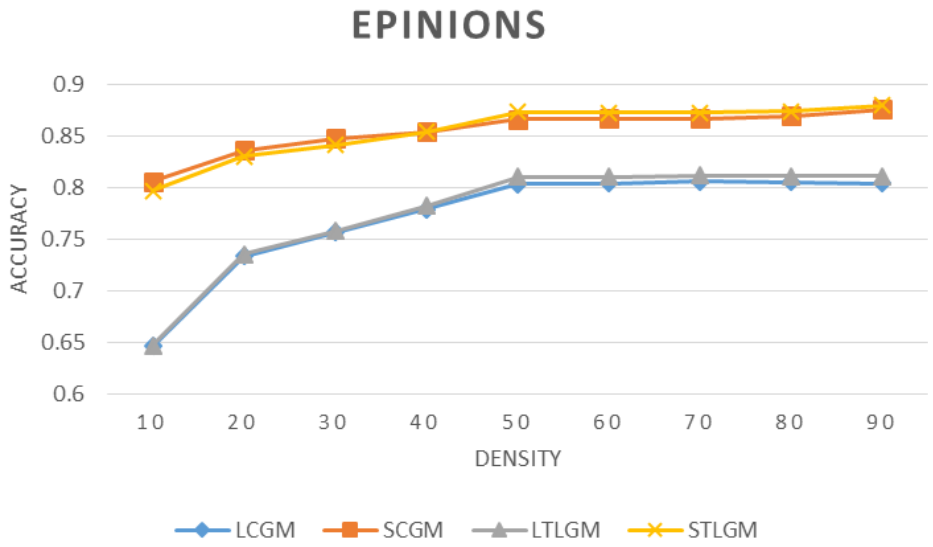

Figure 5.2: Accuracy of the proposed models: LTLGM, LCGM, STLGM, SCGM on networks with various sparsity levels obtained from Epinions dataset.

5.6.4 The Role of Smoothing Model on Networks With Different Levels of Sparsity

In this experiment, we study the effectiveness of the proposed smoothing model on resolving sparsity problem for networks with different levels of sparsity.

To this end, first, we generated networks with various sparsity levels from our original networks by randomly removing edges from them. For example, to obtain a network with 10% density of the original Epinions datasets, 90% of the edges randomly selected and deleted from the network. Note that in this process, nodes were not removed. We applied our prediction models: LTLGM, LCGM, STLGM and SCMG models on the obtained datasets. Fig. 5.2, 5.3 and 5.4 represent the balanced accuracy of the models on networks with various sparsity levels generated from Epinions, Slashdot and WikiElection datasets respectively. As it can be seen, for all three datasets over different sparsity levels, STLGM and SCMG models

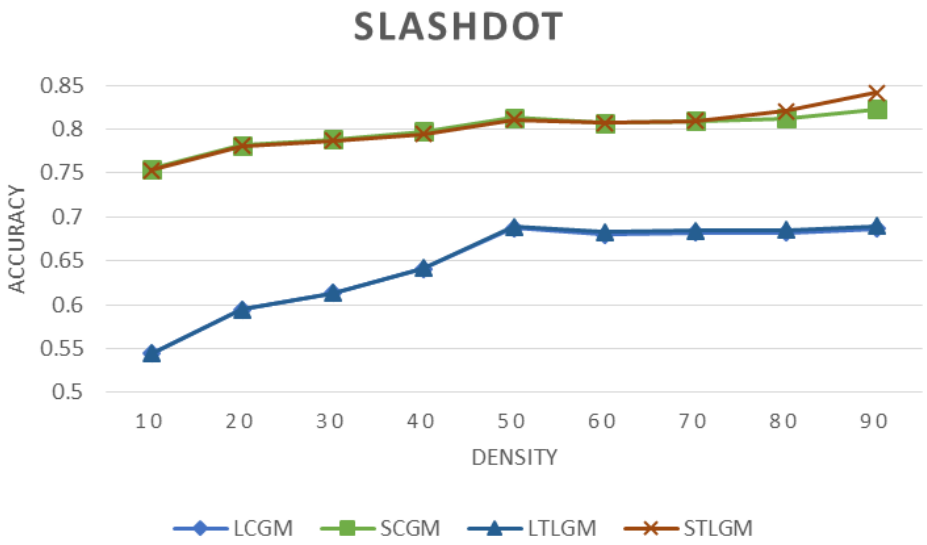

Figure 5.3: Accuracy of the proposed models: LTLGM, LCGM, STLGM and SCMG on networks with various sparsity levels obtained from Slashdot dataset.

substantially improve the accuracy of the local structure-based models. This improvement is more notable on Slashdot and Epinions dataset compared with WikiElection dataset. This can be linked to the structure of the WikiElection dataset where the global structure of the network is not as informative as Epinions and Slashdot datasets. The other observation from this experiment is that as the sparsity of the networks increases, the outperformance of the models with smoothing technique over basic models becomes greater. This observation is quite consistent with our justification for the proposed model. As stated in previous sections, the prediction models based on local information are strongly vulnerable to sparsity problem while global structure has more tolerance toward sparsity. This is the reason that the idea of smoothing is more effective on sparser networks.

5.6.5 Performance of the Proposed Models Versus State-of-the-art Sign Prediction Methods

In this section, we compare the performance of the proposed predictors with six other predictors: those based on social balance theory (BT) and status theory (ST), two models based on machine learning framework: the one introduced by Leskovec et. al. (ML-23) [25] and the extended version proposed by Chiang et. al. (ML-HOC) [163], a model based on Matrix Factorization (MF) [11], a model based on user-based collaborative filtering (CF) [26] and a model based on node embedding (SiNE) [33]. Leskovec et. al. introduced 23 features for sign prediction and used Logistic regression as a classifier. Chiang et. al. extended this

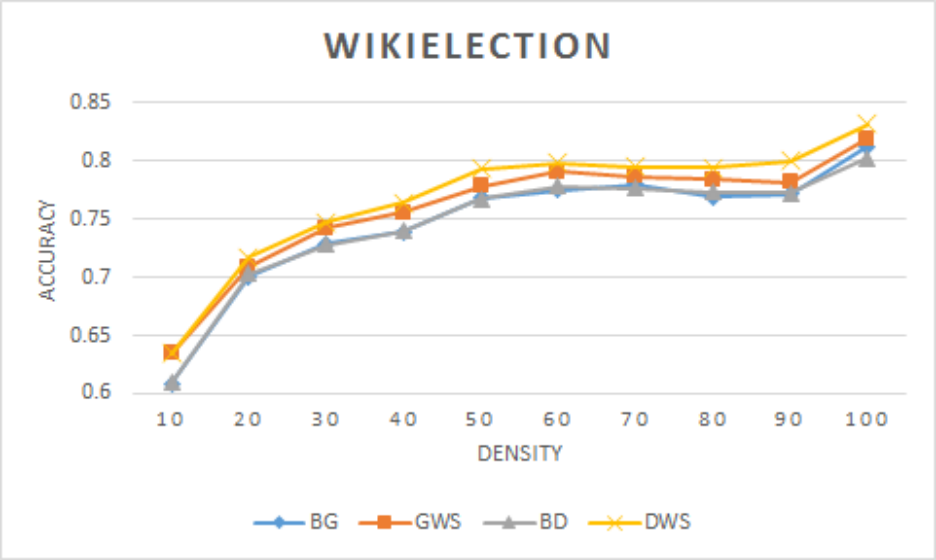

Figure 5.4: Accuracy of the proposed models: LTLGM, LCGM, STLGM and SCMG on networks with various sparsity levels obtained from WikiElection dataset.

work by defining a new set of features extracted from higher order cycles. In our experiments, we extracted the features from cycles of order less than 6, as used in the original work of [163]. The MF-based predictor used in our experiments is the model introduced in [11]. Last, the embedding-based predictor, denoted as SiNE, addresses the prediction task by learning vector representations for nodes through a deep learning framework where the objective function of the model is guided by balance theory [33].

Fig. 5.5 compares the balanced accuracy of the methods on Epinions, Slashdot, and WikiElection datasets respectively. As it can be seen, the smoothing-based models: STLGM and SCGM outperform both the models based on local structures: LTLGM and LCGM as well as those based on global structure: GTLGM and GCGM. Indeed, the accuracies of the individual models are even lower than the state of the art models, while integration of the individual models results in substantial improvements in terms of accuracy. These results shows that how the proposed model can effectively bridge the gap between local and global information sources and leverage them to build a powerful prediction model.

Also, the proposed smoothing-based models presented better accuracy than other state-of-the-art sign prediction methods (i.e., those based on machine learning, social balance and status theories, matrix factorization, embedding and memory-based collaborative filtering). The superior performance of the model can be linked to the information sources used in our proposed model. While the state of the art models mainly rely on either local or global structures, our proposed model in a systematic way combines these two information sources using the idea of smoothing.

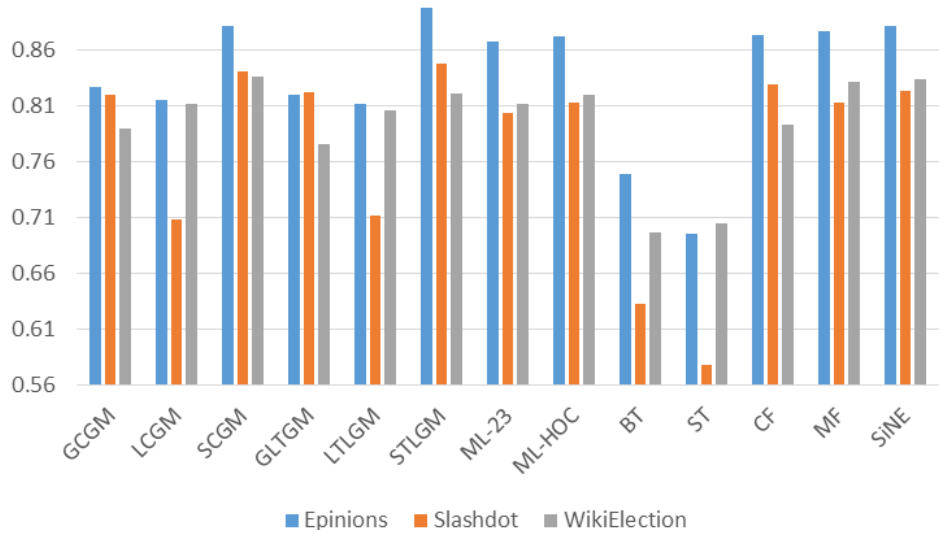

Figure 5.5: Accuracy of Different Predictors Over Epinions, Slashdot and WikiElection Datasets

In all three datasets, predictors based on status and balance theories have the lowest accuracy values. It is beneficial to acknowledge that performances of the social theory-based predictors depend on the meanings of the edges. For example in WikiElection dataset, users vote on each other based on the reliability of the trustee node. Users with higher reliability often receive more positive votes. Reliability of the trustor node can be interpreted as its status. It is the reason why the method based on status theory is a better predictor than the one based on balance theory in this dataset. In Slashdot and Epinions datasets, the votes represent the users' taste and preferences. Therefore, users' voting pattern could be better interpreted based on balance theory than status theory.

The features defined in path-based models also have a basis in these two theories. Similarly, the CF-based and the embedding based predictors also take advantage of the social theories. Therefore, these model these models can not be applied to address the link label prediction problem on networks with more than types of links. However, in our proposed models, we make no assumption about the meaning of the links, and in fact the models can be viewed as a predictor for the general problem of link label prediction.

By comparing the results in terms of consistency on different datasets, it is noticed that the proposed smoothing-based models deliver a consistent performance on all datasets, while this is not the case for other algorithms. We believe that this advantage is due to the fact that the proposed algorithm deals better with the sparsity problem. The proposed smoothing-based models handle this problem by adapting themselves to the sparsity level of the networks. As the network becomes sparser, they give more weight for non-specific reliable approximations

(which come from global structures) and as the reliability of the estimations from local information increases, they put more weight on local information.

5.7 CONCLUSION

In this chapter, we introduced a novel probabilistic approach for sign prediction named as link label modeling. Link label models allow us to find the distribution of possible labels for a target edge. We proposed two link label models based on local structures: LTLGM and LCGM, and two models based on global structures: GTLGM and GCGM. LTLGM and GTLGM models were combined to build another model named as STLGM using the idea of smoothing. Similarly, LCGM and GCGM were combined to build a model named as SCGM. It was shown that the hybrid models could adaptively combine the individual models which allows them to exploit both local and global structures in an intelligent manner. As the sparsity of the input network increases, they rely more on the models based on global structures while they give more weight to the models based on local structures when the sparsity decreases. We evaluated the proposed models on three real-world datasets, and the results showed that the proposed models outperform all previous methods.

The proposed models can be generalized to solve the link classification problem. From application perspective, solving the general problem of link link classification can benefit different personalization problems including recommender systems and intention prediction. For example, a friendship recommendation problem in a social network can be modeled as an interaction prediction problem. Moreover, we can use a link classification model to determine the future interactions of a user in an online system. This can be used to identify the intention behind interactions of the user, which eventually can allow us to personalize the system with respect to the intention of the user [170].

Additionally, the idea of involving content data in our proposed model can be investigated as a future work. For example, content data can be used to better estimate the parameters of the proposed model in both local and global levels. That is, by analyzing the content associated with users and clusters of users we can determine the dependency parameters using content data and mix it with the estimations obtained from graph data. Moreover, our proposed clustering algorithm can also benefit from content data. That is, the cost function of the clustering model can be adjusted in a way that it can take into account both content and graph data.

CHAPTER 6: EMBEDDING SIGNED NETWORKS

6.1 OVERVIEW

Representation learning of graph data obtained from user interactions can be regarded as a comprehensive way for behavioral modeling. In fact, while a link label prediction model is focused on capturing the link labels of the given network, with a relatively higher computation, a representation learning can be designed in a way that it can capture both link structures (the presence or absence of links) as well as link labels. Several applications can be considered for representation learning on networks with multi-types of interactions. For example, a trust relationship among users can be modeled as a signed network [159]. By solving the trust prediction (sign prediction) problem, we can determine what users a certain user trust which can be used to build more powerful collaborative filtering-based recommender systems [159]. Additionally, we can directly use the network of interactions between users for a task like the friend recommendation on networks like Facebook [120]. Also, it is worth mentioning that content data shared by users can also be combined with profiles obtained with graph data to build a more comprehensive model of a user. In fact, this idea has been investigated in some recent works. However, in this thesis, our focus is on challenges associated with the graph data.

While representation learning on graphs with multi-types of edges is of great importance, the vast majority of existing network embedding techniques have been designed for networks having only a single edge type [171]. In classic networks, the relation between two nodes implies closeness. Accordingly, classic embedding techniques try to encode an unsigned network in a way that neighboring nodes are closer in the embedding space [171]. However, in real world networks, links might have more than one type, i.e, a network may contain K types of links where each type represents a different quality of relation between the nodes and as mentioned in the previous chapter, signed networks are an important class of such networks consisting of two types of links: positive and negative [24]. The underlying principles of signed networks can be substantially different from those of classic networks due to existence of both positive and negative links. Therefore, network embedding for signed networks cannot be carried out by simply applying the classic embedding models. While embedding of signed networks is challenging, it has the potential to greatly advance network analysis tasks such as link prediction, sign prediction, and visualization [33] and obviously addressing such tasks has numerous applications associated with behavioral modeling and intention modeling.

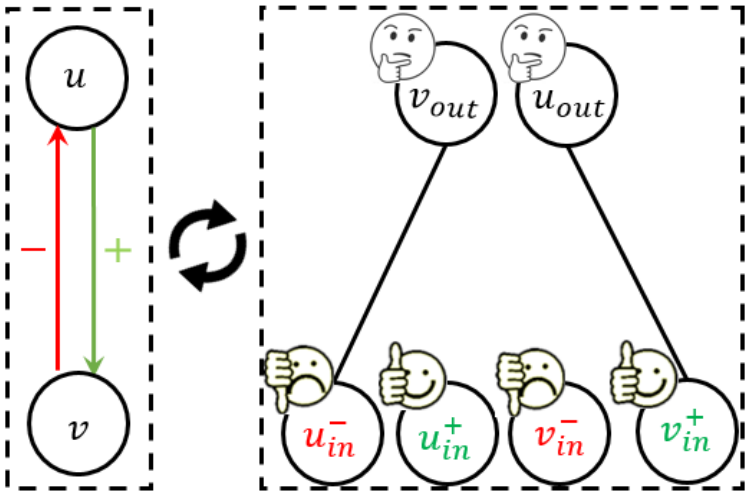

Figure 6.1: Transformation of a signed network with two nodes to an unsigned bipartite network of role-nodes.

Recently, a large body of research has been undertaken to build embedding techniques for signed networks [29, 30, 31, 32]. Analogous to a large class of embedding models in unsigned networks, these models try to embed the network by finding the similarities between nodes based on the paths connecting them. In unsigned networks, a path between two nodes represents their closeness. In fact, embedding based on pat-based similarities allows to capture both **local** and **global** structures in the embedding process. However, defining path-based similarities in signed networks is challenging due to the fact that a signed path may indicate either closeness or distantness. The successful existing works [29, 30, 32] solve this challenge by mainly relying on two social theories that try to explain the formation of signed networks, namely balance theory [172, 173] and status theory [174]. From a broad perspective, these works differ in terms of (1) the maximum path length they use for defining similarities, and (2) the way they define similarity between two nodes with respect to a certain path. However, this general architecture for involving both local and global structures in the embedding process, is associated with two major challenges.

First, social theories are incomplete/inaccurate in explaining signed networks. Both status and balance theories try to explain triad structures in signed networks. However, it has been shown that only around 70% and 65% of triads on a signed network like Epinions comply with status or balance theories [25, 27]. That is, these assumptions have a limited accuracy in explaining the structure of a signed network. Also, it can be said that social theories are more reliable when the network reaches to a stable state. For example, when a set of connections emerge in the network, we cannot expect to see a balanced network immediately. In fact, a balanced structure can be reached over time. However, social networks

are quite dynamic and when we build our user interaction prediction on a certain snapshot of the network, the network is not necessarily in a balanced/stable state. Hence, the models built upon such theories are also affected by the their incompleteness which consequently lowers the quality of embedding techniques.

Second, while classic embedding models aim to capture presence/absence of links, the existing signed embedding models focus only on the labels of connections to generate embedding vectors. In a signed network, three possible interaction *states* can be considered between nodes: absence of a link, a positive link, and a negative link. An embedding model should be able to take into account all these three possible interaction states to fully capture the input network. In other words, a comprehensive embedding technique should enable us to reconstruct both presence/absence of the links as well as the link labels from the embedding vectors. However, the similarity functions of the existing model mainly focus on link labels. As such, their effectiveness in a task like link prediction is limited.

To address these shortcomings, we lay out a new perspective for network embedding denoted as **network transformation-based embedding**: if embedding the underlying network is challenging, it can be transformed to another network for which the embedding task has lower complexity. The transformation can be done by mapping each node in the original network to multiple nodes in the transformed network. Next, the transformed network can be embedded. Finally, the embedding vectors obtained from the transformed network can be aggregated to encode the original network. In fact, for an entity associated with multiple data sources (e.g. text data and graph data), it is a common practice in the literature to first embed the entity with respect to each data source and then aggregate the embeddings [175]. Here we suggest that even when dealing with the graph data as the only data source, we can embed different roles carried by a node separately and eventually combine the vectors.

However, how this general idea can be employed to embed signed networks? We introduce a transformation-based embedding model for signed networks denoted as ROle-based Signed network Embedding (ROSE) that bypasses the aforementioned challenges. An all-to-all connected signed network can be viewed as a bipartite user-item network where each node plays both "user" and "item" roles: item role with respect to connections it receives and user role with respect to links it initiates. Inspired by this analogy, we propose that each node of a signed network can be modeled by a set of roles carried by it, denoted as **role-nodes**, where the relations between role-nodes can be fully captured using *unsigned* links. In this way, the significant novelty is in the transformation of the original network to an unsigned network, which can be fed into any state of the art unsigned embedding technique (e.g., node2vec).

Each role-node captures a certain aspect of a node in the original network. Hence, a comprehensive embedding of a node can be obtained by aggregating the embeddings of the corresponding role-nodes. We introduce two aggregation methods, denoted as *fixed aggregation* and *target-aware aggregation*. A straightforward way to combine the role-node methods is to apply the concatenation function. This model builds a fixed representation for each node. A major application of embedding models is to use embeddings for predicting pairwise interactions, e.g., link prediction. It has been shown that a target aware embedding of entities leads to better results in such applications. For example, a recent deep learning-based recommendation model introduces a model for target dependent encodings of users [176]. Based on this idea, we propose an attention mechanism-based model to aggregate the embeddings of role-nodes in which attendance weights are obtained with respect to the target entity. To the best of our knowledge, this is the first work to build a target aware embeddings of nodes in a signed network.

We evaluated the proposed embedding model based on sign prediction as well as link prediction tasks. The evaluations were carried out on three real-world datasets: Epinions, Wikipedia, and Slashdot. The results of the experiments confirm that the proposed model significantly outperforms all of the existing methods over all of the datasets. Additionally, we show that ROSE has higher flexibility and generalizalibilty compared to the existing models. In summary, the major contribution of the proposed embedding framework is three folds:

1) We introduce for the first time the general idea of network transformation-based network embedding.

2) We propose a novel dedicated network transformation methodology to embed signed networks by transforming them into a bipartite unsigned network that can then utilize any existing state of the art unsigned embedding method to then ultimately obtain node representations for the original signed network.

3) We present the first target aware signed network embedding through our proposed attention mechanisms.

6.2 PRELIMINARIES AND RELATED WORKS

6.2.1 Problem Formulation

In network embedding, the goal is to learn a mapping of each node to a vector space by encoding the link structure of the input network. More formally, the problem can be defined as follows. Let's assume a graph is defined as $G(V, E)$ with a link type mapping function

$\varphi : E \rightarrow A$ where V represents the nodes, E represents the links, and each link $e \in E$ belongs to one particular link type $\varphi(e) \in A$. In unsigned network A has only one values while in signed networks A takes two values: positive and negative. Given the graph G, the task of node encoding is defined as learning a function $f : V \longrightarrow |d|$ that maps each node v to a d-dimensional embedding vector which can be parametrized by the Matrix W with size $|V| * d$.

6.2.2 Unsigned Network Embedding

It has been shown that embedding models can be described as an encoding-decoding framework [28] which consists of four major components: 1) A paiwarise similarity function which measures the similarity of two nodes given a graph G. 2) An encoder function that generates the embedding based on the similarity function. 3) A decoding function which reconstructs the pairwise similarities of two nodes given their embeddings. 4) A loss function, which evaluates how close is the reconstructed similarity values to the true similarity values. Based on this methodology various methods have been introduced in the literature for unsigned networks [12, 43, 177]. Although different embedding techniques rely on different pairwise similarity measures, in all of the works presence of a link between two nodes implies their closeness. [28].

6.2.3 Signed Network Embedding

However, in signed networks two types of relations exists positive and negative [24]. Thus, the similarity functions used in unsigned networks cannot be directly applied to signed networks because the existence of a negative edge does not represent closeness between those nodes. As such, the main challenge in embedding of signed networks is how to involve negative edges in the embedding process without hindering positive proximity.

Recently a large body of research have been conducted to build embedding techniques for signed networks based on similarity functions specifically designed for signed networks. It can be said, all of them try to capture node similarities based on the paths between them [12, 32, 178, 179]. These models can be categorized into three main classes with respect to the length of the paths used for defining similarities: 1) one-length paths, 2) two-length paths and 3) higher order paths.

Single-length paths. A trivial approach to embed signed networks while involving negative links is to embed a node based on its immediate neighbors [175, 178]. For example, in [178], using a cost function, the encoding of nodes is obtained in a way that the similarity

between the embeddings of two nodes is a positive value if a positive link exists between them, and it is a negative value otherwise.

Although it is straightforward to embed a network based on single-length paths, it has a limited effectiveness because it cannot capture the higher order proximities between nodes. In fact, the reason for the success of the unsigned encoding models like node2vec and LINE over the previous works was that they effectively capture higher order proximities between nodes [12, 179]. However, capturing global structures in signed networks is challenging, i.e., given a path containing m positive links and n negative links, how the similarity of the nodes can be determined? Does the path indicate closeness or distantness between the nodes?

Previous works address this challenge with the aid of two social theories: balance theory and status theory. These theories explain the formation of links in signed networks [25].

Balance theory: Balance theory [172, 173] can be described based on the following four rules: "A friend of my friend is my friend," "A friend of my enemy is my enemy," "An enemy of my friend is my enemy," and "An enemy of my enemy is my friend." According to these rules, given a triad only one link with negative link exists in the triad.

Status theory: Status theory [174] in social science states that "the person respected by me should have higher status than me". According to this theory, in signed network, a positive link from node u to v represents the higher status of u than v while a negative link shows the higher status of u.

Paths of length two. As described, social theories originally explain the formation of triangle structures in signed networks. Hence by relying on these theories, given a signed path with length two, it can be determined whether the path indicates closeness or distantness between the target nodes. For example, according to balance theory a path containing a positive and a negative link indicates distantness. A class of recent embedding techniques have attempted to leverage this idea in their embedding process [30, 33]. However, these methods do not go beyond paths of length two. As such, they have limited power in capturing global structures.

Longer paths. Recent works have tried to capture longer cycle paths in their embedding process mainly by relying on the extended version of social theories. [29] runs a random walk on signed networks similar to node2vec algorithm. They suggest that given a random walk-based path between two nodes, the path represents closeness if the number of negative links forming the path is even and it indicate distantness otherwise. The assumption is derived from the generalized version of balance theory and has also been applied to node relevance [180] and personalized ranking [181] using random walks. In another work, again in an attempt to capture higher order proximities, a graph convolutional network method for embedding signed networks has been introduced which relies on balance theory [32]. It

should be noted, [34] is also a random walk-based signed embedding method. Although, it does not rely on social theories but it also makes strong assumptions to define node similarities.

In all, the shared strategy of all these works is that they embed nodes by analyzing the paths between them. If a path indicate closeness they embed the target nodes closer in the embedding space, and distant them otherwise. However, to interpret those complex paths (determining whether a path indicates closeness or distantness) they exploit some strong assumptions which naturally induce noise to the embedding process. Also, this strategy does not use a principled way to distant nodes based on absence of links/paths between them, i.e., it only focuses on capturing positive/negative paths.

6.3 PROPOSED FRAMEWORK: ROSE

In this section, we describe the structure of ROSE. Based on the drawbacks of the previous works, we outline the following **requirements**: an effective universal network embedding model should be able to 1) capture the higher order connectivity between nodes, 2) take into account the link labels as well as the link structures (presence or absence of links), and 3) do not make assumptions about the origin of the network.

Network transformation-based embedding. To address these major requirements, we introduce the general notion of network transformation-based embedding. We suggest that instead of directly finding the path-based similarities of the nodes in the input network, it can be transformed to another network in which we do not encounter the embedding challenges present in the input network. One possible way to do transformation is to define different roles for a node, denoted as *role-nodes* and build a network of role-nodes in a way that the similarities between role-nodes can be determined by adopting the classic similarity functions. Since each role-node captures a certain aspect (role) of an original node, the embedding vector of a target node can be derived by aggregating the embeddings of the corresponding role-node. In sum, a network transformation-based embedding model can be described in three main steps: 1) Network transformation. 2) Embedding the transformed network. 3) Embedding the original network by aggregating the embeddings of the transformed network.

By relying on the general idea of network transformation, we propose ROSE. In the following, ROSE is described based on the aforementioned three-step architecture. We then illustrate how ROSE addresses the requirements of the problem.

6.3.1 Network Transformation

The way the role nodes are defined is fundamental to the effectiveness of the proposed embedding framework. As stated, we aim to define the transformation model in a way that the similarities between role-nodes can be obtained using the classic methods. Note that we do not claim that there is only one single way to define the transformation process. Various embedding techniques have been introduced in the literature based on different similarity measure. Analogously, different embedding techniques can be developed based on the general idea of transformation-based embedding by designing different transformation methods.

Our transformation idea is inspired by an analogy to recommender systems. Traditionally, user-item interactions in the recommender systems context are modeled by a user-item bipartite network. A signed all-to-all connected network can also be viewed as a bipartite network where each node plays a "user" role for the links it creates towards other nodes (given ratings) and plays an "item" role for the links it receives (received ratings). Based on this analogy, we propose that user/item roles of a node can be captured separately through a transformation process and finally combined to build the embedding of the node.

The idea can also be justified from another perspective. The concept behind the mentioned analogy is that we aim to embed the nodes in a way that two nodes are closer in the embedding space if their in-going and out-going linkage patterns are similar. More precisely, two nodes are similar with respect to their outgoing links if they rate *similar* nodes with the same ratings. And two nodes are similar according to their ingoing links if they have been rated by *similar* nodes with the same rating values. This definition is reminiscent of the classic similarity models such as SimRank which states that "two objects are similar if they are referenced by similar objects." [52]. However, it is challenging to capture such similarity function on the original network. Hence, we employ a divide and conquer strategy: each node is mapped to multiple entities (role-nodes) and each entity captures a certain part of the similarity function. In the following the transformation process is introduce more formally.

Transformation to a bipartite network. To separately capture node's in-going and out-going linkage patterns, we map each node to two role-nodes, i.e., given a node u, it is mapped to role-nodes u_{out} and u_{in} where a link from u to v in the original network is modeled as an undirected link between u_{out} and v_{in}. As it can be seen in Fig. 6.2 (a), the input network is transformed to a bipartite network with two types of nodes "in" and "out".

This way, analogous to user-item networks, similarity of role-nodes can be obtained based on their neighboring role-nodes. However, applying a classic similarity measure on the transformed network is still a challenge due to presence of positive and negative links.

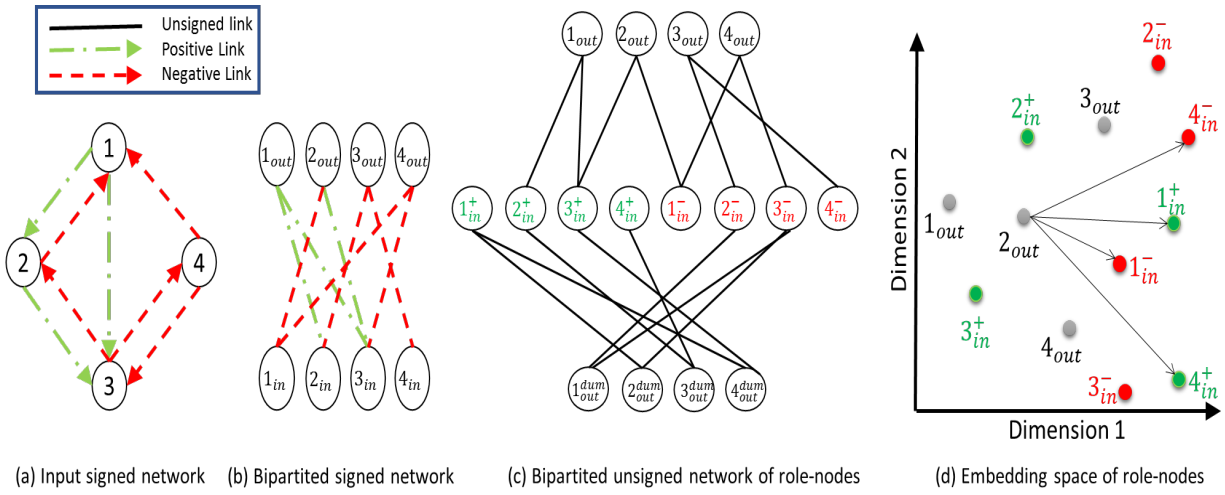

Figure 6.2: Transformation process of role-node.

Transformation to an unsigned network. We transform the network into an unsigned network by defining new role-nodes. A role-node of type "in" v_{in} is mapped into two role-nodes: 1) v_{in}^{+} representing its role when positive links point toward it and v_{in}^{-} representing its role when negative links point towards it. Accordingly, a link from u_{out} to v_{in} with label l is modeled as an unlabeled, undirected link between u_{out} and v_{in}^{l}. In other words, all of the connections between role-nodes become of a single type. This enables us to use the well-established similarity functions to determine the similarities of role-nodes.

Augmenting the network by adding dummy role-nodes. Our strategy is to encode the original network by embedding the transformed network. However, some of the role-nodes may have a very low degree. In particular, role-nodes of type "in-" tend to have a very low degree due to the fact that the number of negative links is often under-represented compared to positive links. According to our results, this can dramatically hinder the accurate embedding of such role-nodes. We further explain this issue in the next subsection.

To address this problem, we aim to inject implicit knowledge about the problem domain into the transformed networks. If node u_{out} has connections towards both v_{in}^{+} and w_{in}^{-}, not only it reflects a similarity between these two role-nodes but also it implies dependence between the opposite role-nodes: v_{in}^{-} and w_{in}^{+}. To bring this important knowledge into our embedding process, which can attenuate the sparsity problem as well, we aim to augment our unsigned network with a set of dummy nodes of type "out". That is, for each node of type "out" in the unsigned network with the set of connections $\{v1_{in}^{l_1}, v2_{in}^{l_2}, ..., vn_{in}^{l_m}\}$, we add a node of type "out-dummy" with the set of connections $\{v1_{in}^{l'_1}, v2_{in}^{l'_2}, ..., vn_{in}^{l'_m}\}$ where l' is the inverse of l, i.e., if l is "in-", l' is "in+" and vice versa.

Summary of transformation. In sum, the input signed graph $G(V, E)$ is transformed

to a bipartite unsigned graph $G_u(V_u, E_u)$ where $|V| = 4|V_u|$, and $|E| = 2|E_u|$, i.e., a node $u \in V$ is mapped to four role-nodes in V_r: 1) role u^{out} which initiates a link, 2) role u^{dummy}_{out} which initiates a dummy link, 3) role u^{+}_{in} which receives a positive link, and 4) role u^{-}_{in} which receives a negative link. And a link $e_{u,v}$ with the label l is transformed to links $e_{u_{out},v^{l}_{in}}$ and $e_{u^{dummy}_{out},v^{l'}_{in}}$. Fig. 6.2 depicts a toy example for the proposed transformation process. It should be noted that the transformation is lossless because the original network $G(V, E)$ can be fully reconstructed from $G_u(V_u, E_u)$.

6.3.2 Embedding the Network of Role-nodes

Analogous to unsigned networks, the links between role-nodes indicates their closeness. Hence, a classic model like node2vec [12] can be used to embed role-nodes. Here we utilize an enhanced node2vec model that we have developed that explicitly takes advantage of the knowledge of our role-nodes through a targeted negative sampling technique. We briefly describe the model in two stages: random walk generation and likelihood optimization.

Random walk generation. The model first generates multiple truncated random walks on the graph. These sequences/paths reveal closeness among role-nodes. At each step, the next node is chosen randomly from the neighbors of the current node. Based on the obtained sequences, pairs of co-occurring role-nodes are extracted from each sequence. A co-occurring pair is defined as two nodes placed within a short distance (a window) in a random walk sequence. It should be noted, a random walk is less likely to visit nodes with low in-degrees. Consequently, such nodes are less likely to appear among co-occurring pairs. This is the reason for augmenting the network by adding dummy nodes in the network transformation phase.

Likelihood optimization. Having the set of co-occurring nodes, the model learns their embedding vectors based on a language model, e.g. skipgram with negative sampling approximation (SGNS). SGNS models the proximity of two nodes as a binary classification over their co-occurrence in a random walk, i.e, the likelihood of m and n appearing as a co-occurring pair in a random walk is defined as:

$$P(m, n) = \sigma(W_m.W_n) = \frac{1}{1 + \exp(-W_m.W_n)}, \tag{6.1}$$

where W_m and W_n are the embedding vectors of m and n, respectively and σ is a Sigmoid function. It implies that a pair of nodes with larger inner product value $W_m \cdot W_n$ has a higher likelihood of co-occurrence. Based on this function, the vector representation of nodes are

learned by optimizing the following cost function:

$$\sum_{(m,n)\in D} [-\log P(m,n) + \sum_{j=1}^{t} -\log(1 - P(m,n'_j))], \tag{6.2}$$

where D is a set of co-occurring node pairs obtained from the first step and n'_j is a set of negative examples with size t. The purpose of negative examples is that it allows the model to penalize 'dissimilar' node pairs of being similar to each other. In our embedding process we use two types of negative examples: random samples and targeted samples.

Random negative sampling. For each co-occurring node pairs, a set of random nodes are sampled from the network. Gradient descent update for noise samples push two randomly sampled nodes apart from each other. Multiple noise pairs are sampled for each co-occurring node pair to account for the imbalance of positive and noise pairs due to the link sparsity.

Targeted negative sampling. A key domain knowledge about the network of role-nodes is that "in+" and "in-" role-nodes of a node carry opposite meanings, i.e, if a node u_{out} is connected to v^+_{in} then it has no connections towards v^-_{in} and vice versa. Hence, it is important to distant the role-nodes representing the positive and negative roles of a node. To this end, we employ the idea of targeted negative sampling. Given a co-occurring example pair, if one of the target node is of type "in" with label l, the opposite role-node (of type "in'_l") is added to the set of negative examples.

6.3.3 Embedding the Original Network

As stated, two nodes in the original network are similar ($u \sim v$) if their corresponding role-nodes are similar ($u_{out} \sim v_{out}$ and $u^l_{in} \sim v^l_{in}$ for all l). Hence, a node's embedding can be obtained by aggregating the embedding vectors of the role-nodes corresponding to the node. In the following we introduce different aggregation models. First model builds a fixed representations of nodes, while the second method aims to build a task/target dependent embeddings of nodes. That is, given a certain entity (e.g. a node), the model builds a representation of the node with respect to the target entity.

Fixed embedding. Based on our proposed technique, a node u in the given signed network is mapped to three main role-nodes: u_{out}, u^+_{in} and u^-_{in}. Note that dummy role-nodes are not used to find the embeddings of the original nodes. As mentioned, the role of dummy nodes is to mitigate the sparsity problem and facilitate finding the similarities of the main roles-nodes. In fact, "out-dummy" role nodes are inverses of the role nodes of type "out"

and do not add extra knowledge about the representations of the original nodes.

Each of the main roles represents a certain perspective/role of a node in the original network. In general, if there are multiple representations of an entity, we can concatenate them or linearly combine them to build a unified representation. Accordingly, a straight forward way to build a comprehensive and unified embedding of a node is to concatenate the embedding vectors of the corresponding role-nodes. As such, the fixed representation of node u can be defined as $W_u = W_{u_{out}}||W_{u^+_{in}}||W_{u^-_{in}}$ in which $||$ represents concatenation.

Target aware embedding. One of the important applications of embedding networks is to use the embedding vectors to predict the pairwise interactions of nodes, e.g., link prediction and link label prediction. Intuitively, when using the encoding of a node to analyze its interactions with another node (or in general an entity), it is more accurate to encode the node with respect to the target entity. We name this target aware profiling/encoding of nodes. To the best of our knowledge, the existing node-embedding techniques build only fixed embeddings.

To embed nodes in a target dependent way, a major question to be answered is how a target entity can be involved in the embedding process of a given node?

The idea of target-aware profiling is the basis for most of the recommendation models. For example, in item-based collaborative filtering, to predict the rating of a user towards an item, her previous ratings are aggregated in a weighted way because not all of the interactions of the user are equally important in reflecting the taste of the user toward the item [182]. Typically, the weight of a rating is determined based on the similarity of the corresponding item to the the target item [176]. As mentioned, our proposed embedding technique relies on an analogy to recommender systems domain. This creates an opportunity to incorporate the ideas used in advanced recommendation models to embed signed networks.

Our method to build target aware embeddings is also inspired by recommender systems. In our embedding framework, intuitively predicting the pairwise interaction from u to v depends on the "out" role node of u (user) and in role nodes of v (item). Analogous to recommender systems, we propose that "out" role-node of u can be embedded according to v which is denoted by $W_{u^v_{out}}$. Having $W_{u^v_{out}}$, the target dependent embedding of u w.r.t. v is defined as $W^v_u = W^v_{u_{out}}||W_u$. Indeed, we concatenate the fixed embedding of u with a component that depends on the target entity to build its target aware embedding.

To build $W_{u^v_{out}}$, we design an attention mechanism-based model. Attention mechanism was originally proposed for machine translation by [183] and since then been applied in many domains such as question answering [184], image classification [185], and more recently in network embedding [40, 186, 187, 188]

We suggest that $W_{u^v_{out}}$ can be obtained by attending to the neighbors of u_{out} based on

their relevancy to the target entity v. More formally, $W_{u^v_{out}}$ is defined as follows:

$$W_{u^v_{out}} = \sum_{s^l_{in} \in N(u_{out})} [e(s^l_{in}, v) W_{s^l_{in}}], \tag{6.3}$$

where $e(s^l_{in}, v)$ is the importance weight of s_{in} w.r.t. v and $N(u_{out})$ is the set of role-nodes connected to u_{out}. (Note that a general technique like node2vec embeds u_{out} by attending equally to all of its neighboring role-nodes.) In the next subsections, we introduce two attention models for estimating importance weights: 1) unsupervised attention and 2) supervised attention.

Unsupervised attention. The intuition behind our unsupervised attention model is that the "in" role-nodes of two nodes are more related if they receive more links from similar nodes. Note that we do not take into account the labels of connections to find the relevancy of two nodes. To systematically obtained the relevance values based on this intuition, we rely on embedding techniques. Given the target signed network, we assume the links are unsigned and transform it to a bipartite network where the obtained network has two types of role-nodes: "in" and "out". Next, the transformed network is embedded using node2vec. Finally, $att(s^l_{in}, v)$ is defined as follows:

$$att(s^l_{in}, v) = \sigma(W_{s_{in}}, W_{v_{in}}) = \frac{1}{1 + exp(-W_{s_{in}}.W_{v_{in}})}. \tag{6.4}$$

In fact, this weight determines how tightly s and v are connected in terms of the nodes that have rated them regardless of the rating values.

Supervised attention. There are some complex patterns helpful for identifying attendance weights that cannot be captured using an unsupervised method. In fact, the unsupervised attention model determines how much the target nodes are related, but it does not measure how much the label of connections towards u an v are correlated. Defining a heuristic to measure such similarities is not straightforward. Moreover, the attention weights obtained from an unsupervised model is not aware of the target task. That is, for a task like link prediction the optimal attention weights could differ from a task like link label prediction.

To extract such complex patterns, we propose a supervised attention model which can be learned in an end-to-end manner based on the final output of the model designed for a certain task. In this way we allow the output of the model to build the similarity function rather than using a predefined similarity function. Fig. 6.3 depicts the structure of the proposed model. To find the attendance weight $att(s^l_{in}, v)$, we concatenate W_s and W_v and

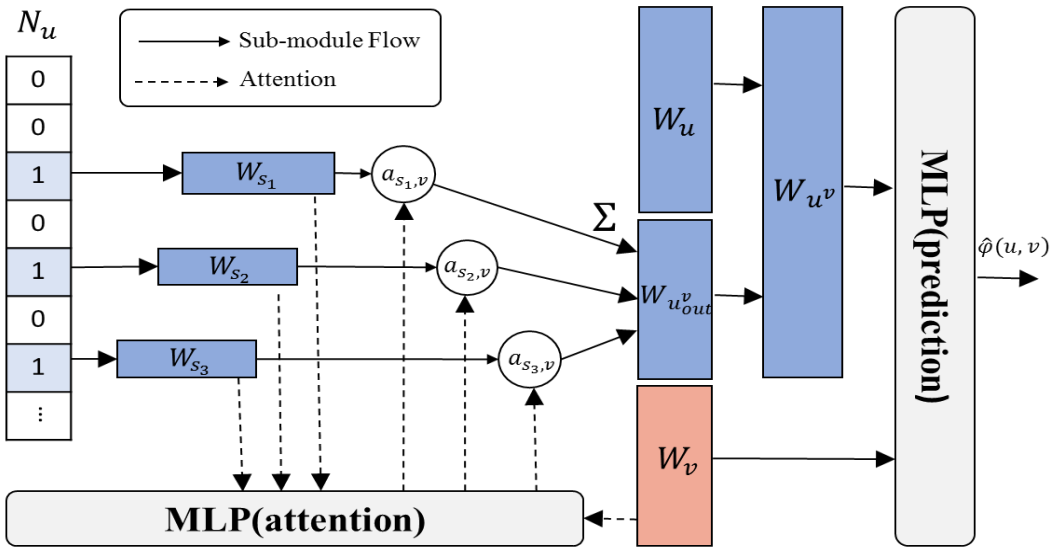

Figure 6.3: Structure of the proposed supervised attention model.

feed them to an MLP. Note that the embeddings of the role-nodes (W_s and W_v) are fixed and do not change during model training. The last layer of the MLP is a Sigmoid function that generates values between 0 to 1. The target attendance weights are obtained by normalizing the outputs of the MLP. Finally, the obtained target aware embedding can be fed to a prediction model for the target task. In fact this way, we generate *target aware* and *task aware* embeddings.

6.3.4 Model Justification

We outline three main requirements for an effective signed embedding technique. The proposed model fulfils those requirements. 1) Unlike the existing models ROSE does not rely on any assumption about the origin of the network. 2) To obtain the embedding of role-nodes we use a random walk-based model similar to node2vec. This implies that the embeddings obtained from role-nodes capture the higher order proximities. 3) The model preserves both link labels and link structures. In addition to addressing the requirements of the problem, the proposed framework creates an avenue to make connection between the recommender systems and signed networks contexts. Based on this idea, we introduce the notion of target aware embedding in signed networks. Lastly, the proposed model is quite generalizable. In the following subsections, we elaborate more on how the model captures link labels and structure and discuss the generalizability of the model.

Preserving link structures and link labels. The embeddings obtained from ROSE can

be used to address the sign prediction and link prediction tasks. To perform link prediction, the embedding vectors of the target nodes can be fed to a nonlinear function trained by a method like MLP where the output of the function is one if a link exist between them and zero otherwise. Analogously, link label prediction can be accomplished using a nonlinear function trained for the task of link label prediction.

Now the question is based on what patterns we expect a non-linear function to address for each of the tasks? As mentioned, the embedding vector of a node consists of three components. A non-linear function can extract various patterns by comparing the different components of the embeddings of the target nodes and combine them to generate the final output.

Major pattern. We encode the role-nodes in a way that if a link with label l exists from u to v, $W_{u_{out}}$ has higher proximity to $W_{v^l_{in}}$ than $W_{v^{l'}_{in}}$. And if there is no link from from u to v, $W_{u_{out}}$ is expected to have low proximities to both $W_{v^l_{in}}$ and $W_{v^{l'}_{in}}$. As such, to predict the existence of a link from u to v, a function can find the similarity of $W_{u_{out}}$ to $W_{v^+_{in}}$ and $W_{v^-_{in}}$. If $W_{u_{out}}$ has low similarity to both $W_{v^+_{in}}$ and $W_{v^-_{in}}$ it indicates there is no link from u to v. To predict the label of the link form u to v, the label of the link is expected to be l if the proximity of $W_{u_{out}}$ to $W_{v^l_{in}}$ is greater than it's proximity to $W_{v^{l'}_{in}}$. In addition to this major pattern, we observe some interesting *implicit* patterns in our experiments. Those patterns are discussed comprehensively in the experiments section.

It is worth mentioning that these patterns allow us to preserve/predict the structure of the network with some level of uncertainty. For example, it is more likely that the label of the edge from u to v is l, if the difference between the proximity of $W_{u_{out}}$ to $W_{v^l_{in}}$ and the proximity of $W_{u_{out}}$ to $W_{v^{l'}_{in}}$ is higher. This way of predicting the network structure is important for certain applications like friend recommendation problem where we aim to confidently recommend a relatively small number of relevant users to a target user.

Model flexibility and generalizability. *Flexibility:* Although, we have adopted a classic embedding technique like node2vec to encode role-nodes, any of the existing models for unsigned networks could be employed. For example, one of the important challenges in signed network is the issue of network sparsity [26, 27]. This problem has been heavily studied in unsinged networks, e.g. , HARP introduces a hierarchical embedding framework to solve the problem [13]. Such methods can be easily applied to our framework to address the sparsity problem in encoding signed network. Graph neural networks (GNN) [41, 189, 190, 191] is another example, with graph convolutional networks (GCN) [192, 193] being a popular type of GNN. A recent work in signed network has tried to to adopt a GCN model to signed networks based on balance theory, SGCN [32]. Similarly, a follow up work of SGCN based on Graph Attention Networks [40] utilized graph motifs to harness both balance and status

theory [186]. However, we propose that with the aid of our proposed framework, all of the existing methods designed for unsinged networks can be easily adopted to networks with multi-types of connections. In other words, there is no need to solve a challenge for signed networks if it has already been solved for unsigned networks.

Generalizability: The fact that the model does not rely on any assumption is beneficial for its generalizability as well. In fact, the model can be easily generalized to address the general problem of link label prediction where the labels of link have more than two classes. To apply the model to a graph with $k>2$ types of labels, we can simply add K role-nodes of type "in" to the transformed network, i.e., a role-node of type "in" is added to the network for each type. In our future works, we will further investigate the generalizability of the proposed framework to other network types, such as knowledge graphs [194], signed bipartite [195, 196] and heterogenous networks [197].

6.4 EXPERIMENTS

We conducted experiments to verify the effectiveness of the proposed framework and the ideas behind the model. The experiments are focused on answering four key questions:

- **RQ1:** How do the proposed embedding frameworks perform when compared to the state of the art models in terms of link-label prediction and link prediction tasks? How effective are the proposed target dependent embedding techniques?
- **RQ2:** What is the interpretation of the embeddings obtained from the network of role-nodes? Furthermore, although we do not utilize signed network social theories, do we observe any similar patters coming from the learned embeddings?
- **RQ3:** Why preserving global structures is important in embedding signed networks?

6.4.1 Experimental Setup

Datasets: We conducted our experiments with three real-world datasets: Epinions, Wiki-Election, and Slashdot which have been used as benchmark in previous works [24]. The statistics of the datasets are provided in Tab. 6.1.

WikiElection: In Wikipedia election, users may give positive or negative votes for the promotion of other users as administrator. WikiElection dataset is the signed network obtained from users' votes for elections of administrators.

Epinions: Epinions was an online product review service. Users could give votes on products and review them. Also, they could express positive or negative votes to other users regarding trustworthiness of their reviews. The Epinions dataset is a signed network that represents the positive/negative relations (votes) between users.

Slashdot: Slashdot dataset is also obtained from an online service (technology news website) where users can share comments and flag each other as friend or foe. The flags indicate approval or disapproval of comments. Analogous to Epinions, Slashdot dataset models the interactions of users using a signed network.

Table 6.1: Datasets statistics

	Nodes	Edges	+Edges	-Edges
Epinions	119217	841200	85.0%	15.0%
Slashdot	82144	549202	77.4%	22.6%
WikiElection	7118	103747	78.7%	21.2%

Evaluation: In our datasets, the number of negative edges is considerably smaller than the number of positive links. As such, comparing prediction methods based on the accuracy of original test sets could be misleading under some circumstances, in particular for the task of sign prediction. Hence, in previous works balanced datasets have been used to do experiments [25, 30]. Following the same methodology, we balanced the datasets by randomly removing positive examples and used 5-fold cross validation to conduct our experiments.

6.4.2 Performance of the Proposed Framework Versus the Baselines on the Sign Prediction and Link Prediction Tasks (RQ1)

In this experiment, we compared the performance of ROSE with four recently introduced signed network embedding models on two tasks: sign prediction and link prediction. Moreover, we compared the target dependent variants of ROSE with the fixed variant on both of the tasks. AUC (Area Under the Curve) was used as the evaluation metric. The following is the list of the models used in this experiment.

SIDE is a random walk-based approach that aims to capture global structures in the embedding process. In fact, the model has adopted the node2vec model for signed networks by relying on balance theory. The model considers the directions of edges in the embedding process [29].

BESIDE is a deep learning approach for encoding signed networks that aims to use both balance and status theories in a complementary manner. In this way, the model leverages

first order and second order proximities to learn embeddings. Similar to SIDE, the model leverages links directions [30].

SiNE is a deep learning-based framework to generate node embeddings in signed networks. The main principle behind the model is that "users should sit closer to their friends (or users with positive links) than their foes". Balance theory is used to develop this idea. It should be noted that the model performs based on undirected networks. In other words, the direction of edges is not taken into account in SiNE [33].

SNE is a random walk-based model that learns representations of nodes by adopting the log-bilinear model and is able to capture both edge types and directions [34].

SIGNet is also a random walk-based model that maintains structural balance using targeted negative sampling. [178].

Furthermore, we evaluate the following three variants of the proposed framework:

ROSE is the basic variant of the model (without using attention) which generates fixed embeddings

ROSE-UAT is the unsupervised attention-based variant of the model which is able to learn target dependent embedding without supervision.

ROSE-SAT is the supervised attention-based variant of ROSE which generates target dependent encodings by employing supervision.

Sign prediction: Sign prediction is the major task that has been used to evaluate encoding models in previous works. In fact, all of the baselines are equipped to a sign prediction model that performs based on their learned node embeddings. The Tab. 6.2 shows the AUC of the models on three datasets. As it can be seen, the basic variant of the model (ROSE) outperforms all of the baselines. For example, ROSE outperforms BESIDE by 0.01%, 0.01% and 0.03% in terms of AUC on WikiElection, Slashdot and Epinions datasets respectively. It should be noted that our results are almost consistent with the AUC values reported in [30]. In fact, BESIDE has the highest accuracy among the counterpart models. The higher accuracy of SINE can be attributed to its effectiveness in addressing the requirements of the problem: 1) It performs based on a similarity function that does not rely on inaccurate assumptions. 2) It captures the global structure of the network in a principled way.

Additionally, we observe that the target dependent variants of the model (ROSE-UAT and ROSE-SAT) perform better than ROSE. In fact, encoding the nodes with respect to a target entity (e.g. a given node) helps to better analyze the interactions of the node and the entity. Also, according to the results adding supervision to the attention mechanism aids it to extract more complex patterns and leverage them in building target dependent representations.

Link prediction: Although link prediction is an important task in network mining,

previous works have not evaluated their models based on link prediction task. To evaluate the models for the link prediction task, we first fed the training graph to the models and obtained the node encodings. Next, we created training and test sets for the task of link prediction based on the obtained node embeddings. Each data instance in the training/test sets is the concatenation of the encoding vectors of a node pair (W_u, W_v) and the label of the instance is 1 if there is a link from u to v and 0 if there is no link. In both training and test sets, 50% of instances have label 1 and 50% of them have label 0. The node pairs with 0 label were randomly selected from the graph. The training set obtained from each embedding method was fed to a Logistic regression and the AUC of the trained model was obtained based on the test set. Tab. 6.2 shows the results of the experiments. As it can be seen, ROSE has superior performance than the baseline models on all of the datasets since the model systematically differentiates the three different interaction-states between nodes in its embedding process. Also, again the target dependent variants of the model outperform the basic variant.

Table 6.2: AUC of the proposed model (ROSE) and the baseline methods on the WikiElection, Slashdot and Epinions datasets.

	Sign Prediction			**Link Prediction**		
Model	**WikiElection**	**Slashdot**	**Epinions**	**WikiElection**	**Slashdot**	**Epinions**
SIDE	0.7986	0.8815	0.8672	0.9184	0.9342	0.9314
BESIDE	0.8953	0.9012	0.9342	0.9092	0.9265	0.9397
SiNE	0.8632	0.8680	0.8543	0.5833	0.5983	0.6488
SIGNET	0.8943	0.8997	0.9181	0.9099	0.8862	0.9205
ROSE	0.9091	0.9082	0.9533	0.9418	0.9357	0.9403
ROSE-UAT	0.9116	0.9095	0.9547	0.9426	**0.9391**	0.9444
ROSE-SAT	**0.9183**	**0.9161**	**0.9568**	**0.9613**	0.9366	**0.9471**

6.4.3 Interpretation of the Encodings Obtained from the Network of Role-nodes (RQ3)

Given nodes u as initiator and v as receiver, three interaction-states can be considered between them: absence of link, positive link and negative link. In previous section, we introduced the major pattern extracted from the distance/similarity of the encoding vectors of u and v that can aid to determine the interaction type between them. In this experiments, we investigated the existence of such patterns.

Fig. 6.4, shows the average distance of different encoding components of a node pair (u, v)

as a function the interaction-state between them. For example, on WikiElection dataset if the link from u to v is positive the average distance of $W_{u_{out}}$ from $W_{v_{in}^+}$ denoted as $d_{avg}(W_{u_{out}}, W_{v_{in}^+})$ is 1.1. As it can be seen, on all of the datasets the average distances of the components are consistent with the patterns introduced in the previous section. The major pattern states that if the link from u to v is positive, $d(W_{u_{out}}, W_{v_{in}^+})$ is expected to be smaller than $d(W_{u_{out}}, W_{v_{in}^-})$. The obtained average distances from the datasets also confirm the existence of such pattern. For example, on Epinions dataset for a positive link, $d_{avg}(W_{u_{out}}, W_{v_{in}^+})$ is 1.1 while $d_{avg}(W_{u_{out}}, W_{v_{in}^-})$ is 1.7. Moreover, we observe that if there is a link from u to v (either positive or negative), $d_{avg}(W_{u_{out}}, W_{v_{in}^-}) + d_{avg}(W_{u_{out}}, W_{v_{in}^+})$ is smaller than the case when there is no link from u to v. For example, on Slashdot dataset, $d_{avg}(W_{u_{out}}, W_{v_{in}^-}) + d_{avg}(W_{u_{out}}, W_{v_{in}^+})$ is 3.2 if there is no link from u to v and it is 2.75 if there is a link. Again this observation approves our introduced major pattern. These pattern are more visible in Fig. 6.5 which depicts the distribution of $d(W_{u_{out}}, W_{v_{in}^-}) - d(W_{u_{out}}, W_{v_{in}^+})$ for different interaction-states.

In addition to the aforementioned major pattern, we observe some other patterns by comparing the distance of the embeddings of role-nodes. We name these patterns as ***implicit patterns*** because our model has not targeted to extract them. In fact, the embeddings of different role-nodes of a node are not statistically independent. As such, to predict the interaction state between two nodes, all of the components of their embedding vectors could be helpful.

1) If the sign of the link from u to v is positive, *similar* nodes rate them similarly and if it is negative *similar* nodes rate them with different signs. These patterns are reflected in the embeddings of the role-nodes of type "in". The smaller distance between the embeddings of the role-nodes of type "in+" and of type "in-" of two nodes means that they were rated by similar nodes similarly. For example, in Epinions dataset $d_{avg}(W_{u_{in}^+}, W_{v_{in}^+})$ and $d_{avg}(W_{u_{in}^-}, W_{v_{in}^-})$ are 1.1 and 1.3 respectively when the sign of the edge from u to v is positive while those distances are 1.7 and 1.6 respectively when the edge sign is negative. Also, in all of the datasets $d_{avg}(W_{u_{in}^+}, W_{v_{in}^-})$ and $d_{avg}(W_{u_{in}^-}, W_{v_{in}^+})$ is smaller when there is a negative link from u to v than when there is a positive link. It can be said this pattern is aligned with balance theory. In fact, the triangle structures described in balance theory can be regarded as a special case of this pattern. As mentioned, balance theory states that if there is a positive (negative) link between two node, their common neighbors develop links of the same (different) sign towards them.

2) In average, u and v rate *similar* nodes more similarly when there is a positive link between them than when there is a negative a link connecting them. The smaller distance values between the embeddings of the role-nodes of type "out" of two nodes indicate that they

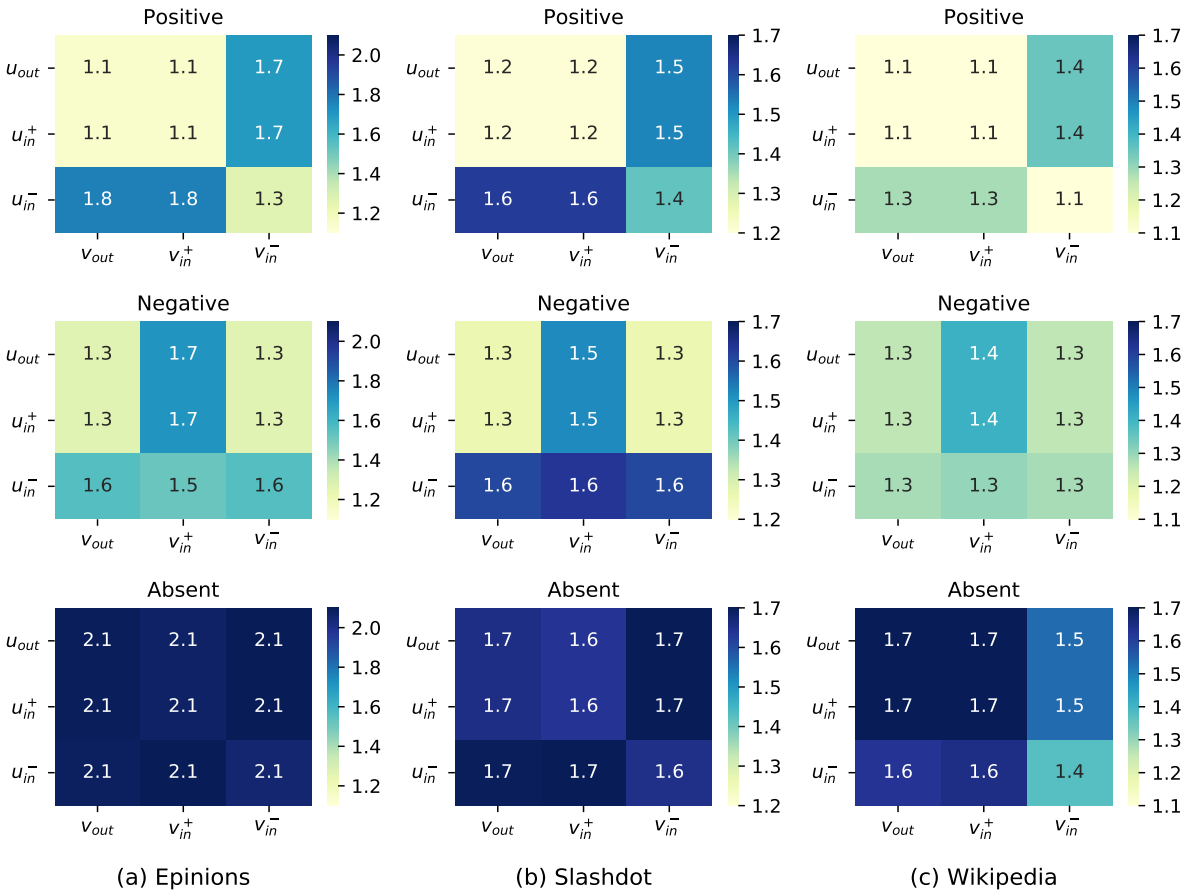

Figure 6.4: The average pairwise distance of the encoding vectors of the role-nodes of a node pair (u, v) for different interaction-types between them: positive link, negative link, and absence of a link.

have rated similar nodes similarly. As it can be seen, for all of the datasets $d_{avg}(W_{u_{out}}, W_{v_{out}})$ is smaller when there is a positive link from u to v. Again, balance theory can be regarded as the special case of this pattern.

3) The signs of the link between two nodes in different directions are correlated. In all of the datasets, $d_{avg}(W_{v_{out}}, W_{u_{in}^+})$ - $d_{avg}(W_{v_{out}}, W_{u_{in}^-})$ is smaller when there is a positive link from u to v than when there is a negative link from u to v. It is worth mentioning that this pattern is contradictory to status theory. According to status theory, the signs of the edges between two nodes in different directions are opposite to each other.

4) The average distance between the embeddings of the role-nodes of two nodes is quite larger when there is no link between them than when there is a link (either positive or negative link). It can be said that a large distance between the embeddings of two nodes implies they are not tightly connected in the network and belong to different clusters.

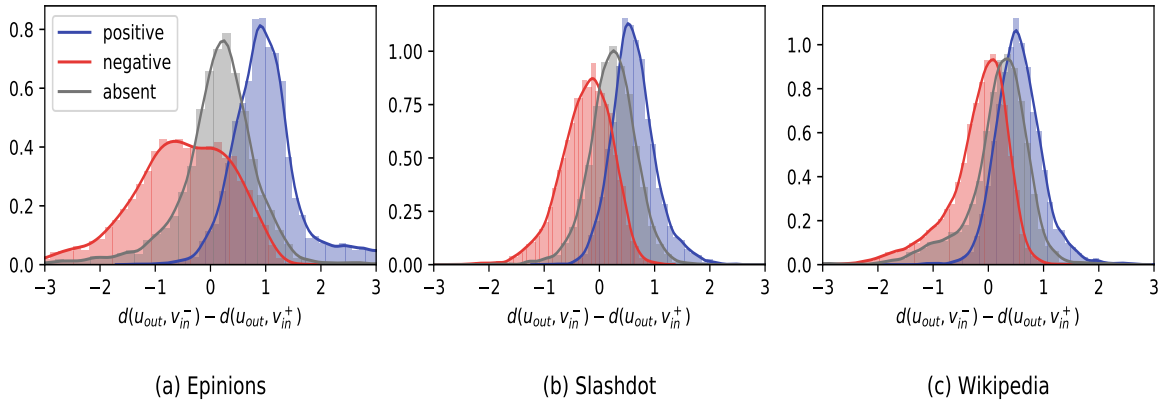

Figure 6.5: The distribution of $d(W_{u_{out}}, W_{v_{in}^+}) - d(W_{u_{out}}, W_{v_{in}^-})$ of a node pair (u, v) for different interaction-types between them: positive link, negative link, and absence of a link.

6.4.4 Effectiveness of Incorporating Global Structures in the Embedding Process (RQ3)

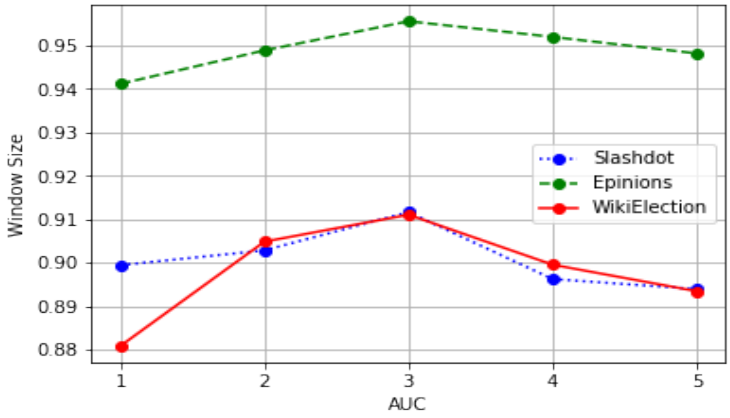

Figure 6.6: AUC of ROSE as a function of window-size parameter.

As stated, one of the major requirements for an effective embedding model is its ability to capture global structures. In ROSE framework, employing random-walks to encode the network of role-nodes enables the model to incorporate higher order proximities in embedding role-nodes as well as original nodes. The window-size parameter in the node2vec model determines the maximum order of proximities used in embedding, i.e., give a random walk if the distance between two role-nodes is larger than the window-size, they are not considered as 'similar' role-nodes. To show the importance of leveraging global-structures in our proposed framework, we evaluated the performance of the model on sign-prediction task as a function of the window-size. As it can be seen in Fig. 6.6, for all of the datasets the best performance was obtained when the window-size was set to 3. It should be noted, the results for link-

prediction task also confirm this conclusion.

6.5 CONCLUSION

In general, existing models for embedding signed networks build their models upon path-based similarity measures where social theories are used to define such similarity measures. However, this structure to build embedding models faces major challenges. 1) Social theories do not accurately explain the structure of signed networks. 2) The models built based on this structure mainly focus on link labels and neglect capturing the link structures (presence/absence of links). In order to address these challenges, we introduced a novel network transformation-based embedding framework denoted as ROSE which relies on transforming the original network to an unsigned bipartite network. To the best of our knowledge, this is the first work to propose the idea of network transformation-based embedding. Our experiments confirmed that the model outperforms the state of the art models on sign prediction and link prediction tasks. Moreover, the model is generalizable to networks with more than two types of connections. Also, it is capable of encoding nodes with respect to a target entity. We plan to investigate ideas of how to use network transformation for other graph types, e.g., knowledge graphs [194], so that they can also benefit from the state-of-the-art embedding methods being developed for simple graphs. Another direction is to improve other signed network tasks using the embeddings coming from ROSE, e.g., community detection [198, 199] and node ranking [30, 200, 201].

The proposed model and its possible extensions has various applications. Recommender systems can benefit from the model in both direct and indirect ways. In a social recommendation-based setting, we can use the embeddings obtained from a networks of interactions to find neighboring users of a target user which can boost the performance of classic recommender systems [95]. Moreover, we can directly use the obtained embeddings to address the user-to-user recommendation task [202]. Also, it should be mentioned that we can even view user-item interactions as a bipartite network with multi-types of edges. Considering the generalizability of the proposed technique, we can use it to profile users in such networks and make recommendations. Lastly, the proposed model can be easily extended to leverage content data. A trivial solution is to build independent graph-based and content-based embeddings of users and then combine them using a classic technique like concatenation. In a more systematic way, we can build a graph of relationships between users based on their content data, i.e., a type of connection between two users can be determined by comparing their content data. The original network can be augmented based on the obtained connections. Finally, we can perform representation learning on the obtained network.

CHAPTER 7: CONCLUSION AND RESEARCH FRONTIERS

In this thesis, we have aimed to solve problems related to user modeling on social networks based on the graph data, where the main focus has been on integrating local and global structures of the target graphs. We consider two main classes for user profiling, denoted as interest modeling and behavioral modeling, and introduced problems representing each of the classes. We have shown that for our proposed interest modeling-based problems, the existing models face the *efficiency* challenge. Moreover, we have shown that the state-of-the-art models for behavior modeling encounter the *complexity* challenge. In response, we propose general concepts and ideas that address these challenges systematically.

7.1 CONCLUSION

In this thesis, we have made contributions in the following aspects to integrating local and global structures for user profiling purposes.

1. We have proposed user modeling problems in interest-based networks that require us to build models that can bridge local and global structures with high efficiency: interest-based user discovery and recommendation solely based on graph data. The problems have been introduced in Chapters 3 and 4, respectively.

Interest-based user discovery has important applications in different settings. In this problem, the goal is to identify users in the network who share interest patterns similar to a given seed set of users. Considering the nature of the problem, it is a requirement of the problem to exploit global structures to address the problem while maintaining high efficiency. However, a model developed based on the existing techniques like graph neural networks needs to analyze the entire network in multiple iterations. Clearly, such operations have burdensome computational costs, which does not comply with the requirements of the problem.

In the problem of recommendation based on graph data, given a seed/training set of users associated with a set of items, the goal is to identify the relevance of the rest of the users/nodes in the network to the target items. Again, in this problem, if we rely on existing techniques, we need to examine the entire network. However, considering the dynamic of the network and the dynamic of the set of items, such approaches are not suitable for the problem, mainly due to their high computational costs. In general, if we have a profiling problem where the entire set of users within a network need to be modeled based on a small set of *labels* users, it is key to involve both local and global structures. However, doing so

based on the state-of-the-art models in an efficient manner is challenging.

2. We have proposed that in order to build a user modeling technique that relies on integrating local and global structures, it is beneficial to take advantage of the network generation process. In fact, it has been shown that for some types of networks, in particular scale-free networks, the link generation process in local structures can determine the organization of global structures [19]. For example, in a network where users tend to make connections towards popular users, the network is expected to have a star shape structure (high scale-freeness). We suggest that the network generation process can also hint at how to integrate local and global structures for a user profiling task. In interest-based networks like Twitter network, for each topic of interest, there exist a set of hub-nodes, and users interested in a topic tend to make connections to the hub nodes associated with the topic. In fact, such networks consist of star-shaped clusters where each cluster is characterized by a set of hub nodes that receive connections from users/nodes belonging to that cluster.

Considering the link generation process of interest-based networks, our idea for user profiling involves profiling hub nodes of a network given a seed set of users. We suggest that given a training/seed set of users in a network, we can obtain the profiles of the cluster centers of the network, which allows us to determine the cluster structure of the network. This can be viewed as a method for clustering scale-free networks and profiling each cluster. Next, knowing that users tend to be directly connected to cluster centers, we can use users' local structures to profile them, i.e., a user can be modeled based on the cluster centers it is connected to. Unlike existing models like GNNs, the model is quite efficient and can handle network dynamics.

3. In Chapters 5 and 6, we have investigated the complexity of user modeling based on both local and global structures in networks representing multi-types of interactions. Various applications can be considered for a user model obtained from a network of interactions including recommender systems. For example, using the idea of social recommendation, by building the interaction-based profiles of users, we can identify neighbors of a users. Based on the notion of homophily, the interest-based profiles of the neighboring users of a given user can be aggregated to build a more accurate profile of the user. Trust prediction, anomaly detection and user-to-user recommendation are other applications that can be benefit from an interaction modeling method.

Several successful techniques have been introduced for user modeling by leveraging both local and global structures in interaction-based networks where the interactions have a single type [22]. However, generalizing such works to networks with multi-type of interaction is challenging. We have focused on signed networks as a representative of interaction-based networks with multi-type of interactions and investigated two major problems: edge sign

prediction and representation learning. We have shown that integrating local and global structures for user modeling to address the problems is challenging. In fact, this challenge stems from the fact that defining the notion of proximity between users in networks with multi-type of interactions. In a network with a single type of interaction, weaker ties between two nodes represent their lower proximity. However, that is not necessarily the case in networks with multi-types of connections. As a result, defining informative clusters for user profiling and involving higher-order paths in the profiling process becomes a challenge.

4. We have proposed that the complexity of involving global structures of networks representing multi-types of interactions can be resolved by defining a customized notion of nodes proximity and graph clusters in such networks. As mentioned, cluster structures and proximity measures used in classic networks are not applicable in networks with multi-type of interactions. For the task of edge label prediction, we introduce a probabilistic model that relies on a novel clustering technique for signed networks. And for the task of representation learning, we have introduced a novel framework for analyzing networks with multi-types of interaction denoted as transformation-based representation learning. The core idea behind the framework involves transforming a given network with multi-type interactions into a network with a single type of interaction. Having the transformed network, any state-of-the-art model designed for a classic network that leverages both local and global structures can be used for representation learning. Finally, the obtained representations from the transformed network can be integrated to represent the nodes in the original networks. We have shown that this general idea can effectively and efficiently solve the problem.

7.2 RESEARCH FRONTIERS

There are many research ideas based on the proposed concepts that can be explored in future works. Here we illustrate a few of them.

1. **Bridging local and global structures based on the network generation process.** A large body of research has been conducted about how the network generation process in local structures can form global structures [19]. For example, it has been shown that if the nodes of a network tend to make connections towards high-degree nodes, it results in high scale-freeness [17]. We have used this important connection between local and global structures in scale-free networks to build more efficient user modeling techniques in such networks. However, this general idea can be employed to build more effective/efficient user modeling techniques, even for other network types.

2. **Hierarchical clustering in interest-based networks based on the notion of representative/hub nodes.** We have introduced the notion of hub nodes in interest-based

networks, which can be viewed as the cluster centers of a given network, and suggested that users can be profiled based on cluster centers they are directly connected to. However, we believe that the potentials of this general idea can be further exploited if the hierarchy of cluster structures can be determined. That is, the cluster centers of a given network can be organized into a hierarchy based on how general the topics they represent are. This way, by analyzing the connections of users towards the hub nodes, a clearer map of the user's interests can be obtained. In fact, this allows us to build user profiles in different granularities. For example, it can be determined a user is interested in politics, in particular the politics of the USA, and she is a Democrat. This hierarchical structure for cluster centers can be seen as a knowledge graph. By exploiting this knowledge graph, various questions can be answered about a target user. That is, the existing models for question answering based on knowledge graphs can be applied to the obtained network.

3. **Extending the idea of network transformation-based representation learning to knowledge graph.** We have introduced the idea of network transformation for representation learning based on shallow embedding techniques for signed networks. However, this idea can be extended in different aspects. Firstly, the model can be applied to a more general setting like knowledge graphs or the network of user-item interactions. Considering the fact that our proposed model is not specific for signed networks, it can be easily applied to such settings. This way, deep and complex interactions can be extracted from knowledge graphs. Moreover, the recommender systems domain can also benefit from the idea of transformation. For example, consider a bipartite network of user-item interactions that captures different types of interactions that users may have with items, like visiting items, adding them to favorites, and purchasing them. It is quite beneficial for a recommender system to differentiate these types of interactions and leverage all of them to build a recommender system. The idea of network transformation allows to differentiate these interaction types and extract multi-step patterns from users' behaviors which can be used to make more informed recommendations. For example, by involving longer cycle paths in the profiling/recommendation process, it can be determined if a user adds the item A as favorites, which item is more likely to be purchased by the user. Or if a user has purchased the item A, what items should not be recommended to the user anymore.

Also, the idea can be further extended to equip graph convolutional networks to the idea of network transformation. In fact, the extended versions of GCN model can perform on networks with multi-types of interactions [22]. However, they need to build representations for each interaction type which increases the complexity of such models. With the aid of the idea of network transformation, the classic GCN models can be directly applied to networks with multi-types of interaction without the need to learn additional parameters.

4. **Generalizing the idea of transformation-based representation learning by learning how to transform.** In our proposed idea for network transformation, the algorithm to transform an original network into a transformed network is of great importance. Our representation learning algorithm designed for signed networks relies on a transformation method specifically designed for signed networks. However, we suggest that in order to further strengthen the generalization ability of the proposed framework, the transformation process can be learned in an end-to-end manner based on the final output of the systems. That is, given a network with K interaction types , and having M node-states for each node, the model can learn how a certain interaction type can be mapped to connections towards the node-states.

5. **Applying the proposed models to different personalization settings.** We investigated user profiling problems on both interest-based and interaction-based networks. The core idea of the models proposed to address the introduced problems can be applied to different personalization settings.

The hashtag recommendation model proposed in Chapter 3 can be viewed as a general interest profiling model that can be applied to various recommendation problems (not limited to the hashtag recommendation problem). Moreover, we can use the profile obtained from graph data to better understand the content generated by users [203]. For example, the graph-based profile can help to better analyze the intent behind search queries generated by users, which consequently can improve the quality of search results retrieved by the system [204]. Also, as mentioned, the proposed model can also be viewed as a clustering technique which indeed provides profiles for each cluster. From this perspective, we can employ the model as a visualization technique, i.e., we can represent each cluster based on the keywords or hashtag associated with their cluster centers.

The user discovery model proposed in Chapter 4 can be used in various applications. For example, with some adjustments, the proposed model can be employed to develop a targeted advertisement model or a polling system to discover users holding a certain opinion about a certain topic. Also, using the proposed idea, we can define some novel problems in finding proximities between users. Most of the existing techniques for defining proximities between users return general similarity measures [171, 205]. However, user similarities can be defined in an aspect-aware manner, i.e., how similar are two users with respect to a certain topic. The core idea of our discovery system can be employed to address this problem. That is, given a certain aspect represented by a seed set of users, we can determine how much two users are similar. This idea is more helpful if we aim to build a user-to-user recommender system in social networks where users are recommended based on certain shared interests. Additionally, it can help us to make explanations along with our recommendations [206].

The representation learning and behavioral prediction techniques introduced in Chapters 5 and 6 can also be used in different personalization settings. Specifically, the models developed for signed networks can be directly used as trust prediction methods that have application in trust-aware recommender systems [159]. In a more general setting, modeling users in a network of interactions can be used to build social interaction-aware recommender systems [102]. Additionally, user-to-user recommender systems can rely on interaction modeling techniques.

6. **Extending the proposed models to involve other data sources.** While in this thesis we have focused on challenges associated with user profiling based on graph data, the proposed techniques can be extended to involve other data sources associated with users, in particular content data. Roughly speaking, adding more 'informative' data sources in the profiling process can lead to building more accurate profiles of users [207, 208]. For example, we have shown that the graph-based hashtag recommendation model introduced in Chapter 3 can be extended to leverage content data. Our results show that the hybrid model outperforms the graph-based and content-based models. Similarly, we can extend the interest-based user discovery system introduced in Chapter 4 to take advantage of content data. For example, rather than defining a target interest topics based on only hub nodes, we can define them based on both hub nodes and words. That is, we can extend the proposed query-driven topic modeling algorithm in a way that it can generate the tweets of users as well as their connections towards hub nodes. Involving more *signals* in the profiling process not only helps us to discover more accurate topics but also it allows to better determine the relevance of users to each of the target topics.

The proposed models in chapters 5 and 6 can also benefit from content data. As mentioned, the probabilistic model introduced in Chapter 5 can exploit content data to better estimate the parameters of the model. Moreover, content data can be leveraged to better cluster signed networks, i.e., we can adjust the cost function of the proposed clustering algorithm in a way that it can take into account the similarity of the content shared by users as well as the similarity of the linkage patters. As an alternative idea, the content shared by users can be used to extract another layer of connections between users, i.e., we can determine positive and negative relations between users based on the content data associated with them. By combing the obtained graph with the original graph, we can build more comprehensive profiles of users. This idea of augmenting the original graph based on content can be injected into the representation learning model introduced in Chapter 6 as well.

REFERENCES

[1] P. Brusilovski, A. Kobsa, and W. Nejdl, *The adaptive web: methods and strategies of web personalization*. Springer Science & Business Media, 2007, vol. 4321.

[2] M. Gao, K. Liu, and Z. Wu, "Personalisation in web computing and informatics: Theories, techniques, applications, and future research," *Information Systems Frontiers*, vol. 12, no. 5, pp. 607–629, 2010.

[3] G. Piao and J. G. Breslin, "Inferring user interests in microblogging social networks: a survey," *User Modeling and User-Adapted Interaction*, vol. 28, no. 3, pp. 277–329, 2018.

[4] W. Gong, E.-P. Lim, and F. Zhu, "Characterizing silent users in social media communities," in *Ninth International AAAI Conference on Web and Social Media*, 2015.

[5] N. Barbieri, F. Bonchi, and G. Manco, "Who to follow and why: link prediction with explanations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 1266–1275.

[6] M. McPherson, L. Smith-Lovin, and J. M. Cook, "Birds of a feather: Homophily in social networks," *Annual review of sociology*, vol. 27, no. 1, pp. 415–444, 2001.

[7] H. Kwak, C. Lee, H. Park, and S. Moon, "What is twitter, a social network or a news media?" in *Proceedings of the 19th international conference on World wide web*. AcM, 2010, pp. 591–600.

[8] G. Farnadi, J. Tang, M. De Cock, and M.-F. Moens, "User profiling through deep multimodal fusion," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, 2018, pp. 171–179.

[9] Z. Li, B. Guo, Y. Sun, Z. Wang, L. Wang, and Z. Yu, "An attention-based user profiling model by leveraging multi-modal social media contents," in *Cyberspace Data and Intelligence, and Cyber-Living, Syndrome, and Health*. Springer, 2019, pp. 272–284.

[10] R. Li, C. Wang, and K. C.-C. Chang, "User profiling in an ego network: co-profiling attributes and relationships," in *Proceedings of the 23rd international conference on World wide web*, 2014, pp. 819–830.

[11] K.-Y. Chiang, C.-J. Hsieh, N. Natarajan, I. S. Dhillon, and A. Tewari, "Prediction and clustering in signed networks: a local to global perspective," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1177–1213, 2014.

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

[13] H. Chen, B. Perozzi, Y. Hu, and S. Skiena, "Harp: Hierarchical representation learning for networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[14] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[15] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *arXiv preprint arXiv:1901.00596*, 2019.

[16] J. Leskovec, D. Huttenlocher, and J. Kleinberg, "Signed networks in social media," in *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 2010, pp. 1361–1370.

[17] A.-L. Barabási and E. Bonabeau, "Scale-free networks," *Scientific american*, vol. 288, no. 5, pp. 60–69, 2003.

[18] S. Aparicio, J. Villazón-Terrazas, and G. Álvarez, "A model for scale-free networks: application to twitter," *Entropy*, vol. 17, no. 8, pp. 5848–5867, 2015.

[19] G. Robins, P. Pattison, and J. Woolcock, "Small and other worlds: Global network structures from local processes," *American Journal of Sociology*, vol. 110, no. 4, pp. 894–936, 2005.

[20] S. Y. Jung, J.-H. Hong, and T.-S. Kim, "A statistical model for user preference," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 834–843, 2005.

[21] A. Javari, Z. He, Z. Huang, R. Jeetu, and K. Chen-Chuan Chang, "Weakly supervised attention for hashtag recommendation using graph data," in *Proceedings of The Web Conference 2020*, 2020, pp. 1038–1048.

[22] W. L. Hamilton, "Graph representation learning," *Synthesis Lectures on Artifical Intelligence and Machine Learning*, vol. 14, no. 3, pp. 1–159, 2020.

[23] T. Derr, "Network analysis with negative links," in *Proceedings of the 13th International Conference on Web Search and Data Mining*, ser. WSDM '20, 2020.

[24] J. Tang, Y. Chang, C. Aggarwal, and H. Liu, "A survey of signed network mining in social media," *ACM Computing Surveys (CSUR)*, vol. 49, no. 3, p. 42, 2016.

[25] J. Leskovec, D. Huttenlocher, and J. Kleinberg, "Predicting positive and negative links in online social networks," in *Proceedings of the 19th international conference on World wide web*. ACM, 2010, pp. 641–650.

[26] A. Javari and M. Jalili, "Cluster-based collaborative filtering for sign prediction in social networks with positive and negative links," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 5, no. 2, pp. 1–19, 2014.

[27] A. Javari, H. Qiu, E. Barzegaran, M. Jalili, and K. C.-C. Chang, "Statistical link label modeling for sign prediction: Smoothing sparsity by joining local and global information," in *2017 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2017, pp. 1039–1044.

[28] W. L. Hamilton, R. Ying, and J. Leskovec, "Representation learning on graphs: Methods and applications," *arXiv preprint arXiv:1709.05584*, 2017.

[29] J. Kim, H. Park, J.-E. Lee, and U. Kang, "Side: Representation learning in signed directed networks," in *Proceedings of the 2018 World Wide Web Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2018, pp. 509–518.

[30] Y. Chen, T. Qian, H. Liu, and K. Sun, "Bridge: Enhanced signed directed network embedding," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 2018, pp. 773–782.

[31] S. Wang, C. Aggarwal, J. Tang, and H. Liu, "Attributed signed network embedding," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 2017, pp. 137–146.

[32] T. Derr, Y. Ma, and J. Tang, "Signed graph convolutional networks," in *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2018, pp. 929–934.

[33] S. Wang, J. Tang, C. Aggarwal, Y. Chang, and H. Liu, "Signed network embedding in social media," in *Proceedings of the 2017 SIAM international conference on data mining*. SIAM, 2017, pp. 327–335.

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

[35] A. Javari, T. Derr, P. Esmailian, J. Tang, and K. C.-C. Chang, "Rose: Role-based signed network embedding," in *Proceedings of The Web Conference 2020*, 2020, pp. 2782–2788.

[36] S. Abu-El-Haija, B. Perozzi, A. Kapoor, N. Alipourfard, K. Lerman, H. Harutyunyan, G. Ver Steeg, and A. Galstyan, "Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing," in *international conference on machine learning*. PMLR, 2019, pp. 21–29.

[37] H. Chen, B. Perozzi, R. Al-Rfou, and S. Skiena, "A tutorial on network embeddings," *arXiv preprint arXiv:1808.02590*, 2018.

[38] M. Chen, Z. Wei, Z. Huang, B. Ding, and Y. Li, "Simple and deep graph convolutional networks," in *International Conference on Machine Learning*. PMLR, 2020, pp. 1725–1735.

[39] C. Yuan, Q. Ma, W. Zhou, J. Han, and S. Hu, "Jointly embedding the local and global relations of heterogeneous graph for rumor detection," in *2019 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2019, pp. 796–805.

[40] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[41] M. Niepert, M. Ahmed, and K. Kutzkov, "Learning convolutional neural networks for graphs," in *International conference on machine learning*, 2016, pp. 2014–2023.

[42] F. Aziz, H. Gul, I. Uddin, and G. V. Gkoutos, "Path-based extensions of local link prediction methods for complex networks," *Scientific reports*, vol. 10, no. 1, pp. 1–11, 2020.

[43] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 1225–1234.

[44] M. Jamali and M. Ester, "Trustwalker: a random walk model for combining trust-based and item-based recommendation," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009, pp. 397–406.

[45] W.-L. Chiang, X. Liu, S. Si, Y. Li, S. Bengio, and C.-J. Hsieh, "Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 257–266.

[46] F. Li, J. He, G. Huang, Y. Zhang, and Y. Shi, "Retracted: A clustering-based link prediction method in social networks," 2014.

[47] L. Lü and T. Zhou, "Link prediction in complex networks: A survey," *Physica A: statistical mechanics and its applications*, vol. 390, no. 6, pp. 1150–1170, 2011.

[48] T. Zhou, L. Lü, and Y.-C. Zhang, "Predicting missing links via local information," *The European Physical Journal B*, vol. 71, no. 4, pp. 623–630, 2009.

[49] Y. Pan, D.-H. Li, J.-G. Liu, and J.-Z. Liang, "Detecting community structure in complex networks via node similarity," *Physica A: Statistical Mechanics and its Applications*, vol. 389, no. 14, pp. 2849–2857, 2010.

[50] E. A. Leicht, P. Holme, and M. E. Newman, "Vertex similarity in networks," *Physical Review E*, vol. 73, no. 2, p. 026120, 2006.

[51] L. Katz, "A new status index derived from sociometric analysis," *Psychometrika*, vol. 18, no. 1, pp. 39–43, 1953.

[52] G. Jeh and J. Widom, "Simrank: a measure of structural-context similarity," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2002, pp. 538–543.

[53] P. Chebotarev and E. Shamis, "The matrix-forest theorem and measuring relations in small social groups," *arXiv preprint math/0602070*, 2006.

[54] F. Fouss, A. Pirotte, J.-M. Renders, and M. Saerens, "Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation," *IEEE Transactions on knowledge and data engineering*, vol. 19, no. 3, pp. 355–369, 2007.

[55] H. Cai, V. W. Zheng, and K. C.-C. Chang, "A comprehensive survey of graph embedding: Problems, techniques, and applications," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 9, pp. 1616–1637, 2018.

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

[57] D. Zhu, P. Cui, Z. Zhang, J. Pei, and W. Zhu, "High-order proximity preserved embedding for dynamic networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 11, pp. 2134–2144, 2018.

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

[59] S. Zhang, H. Tong, J. Xu, and R. Maciejewski, "Graph convolutional networks: a comprehensive review," *Computational Social Networks*, vol. 6, no. 1, pp. 1–23, 2019.

[60] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE transactions on neural networks and learning systems*, 2020.

[61] Y. Feng, H. You, Z. Zhang, R. Ji, and Y. Gao, "Hypergraph neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 3558–3565.

[62] J. Jiang, Y. Wei, Y. Feng, J. Cao, and Y. Gao, "Dynamic hypergraph neural networks." in *IJCAI*, 2019, pp. 2635–2641.

[63] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European conference on computer vision*. Springer, 2016, pp. 630–645.

[64] L. Chen, L. Wu, R. Hong, K. Zhang, and M. Wang, "Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, 2020, pp. 27–34.

[65] Y. Rong, W. Huang, T. Xu, and J. Huang, "Dropedge: Towards deep graph convolutional networks on node classification," *arXiv preprint arXiv:1907.10903*, 2019.

[66] J. Gallier, "Spectral theory of unsigned and signed graphs. applications to graph clustering: a survey," *arXiv preprint arXiv:1601.04692*, 2016.

[67] C. C. Aggarwal and H. Wang, "A survey of clustering algorithms for graph data," in *Managing and mining graph data*. Springer, 2010, pp. 275–301.

[68] X. Wang, D. He, D. Chen, and J. Xu, "Clustering-based collaborative filtering for link prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 29, no. 1, 2015.

[69] E. Faerman, F. Borutta, K. Fountoulakis, and M. W. Mahoney, "Lasagne: Locality and structure aware graph node embedding," in *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*. IEEE, 2018, pp. 246–253.

[70] F. Borutta, J. Busch, E. Faerman, A. Klink, and M. Schubert, "Structural graph representations based on multiscale local network topologies," in *IEEE/WIC/ACM International Conference on Web Intelligence*, 2019, pp. 91–98.

[71] X. Wang, P. Cui, J. Wang, J. Pei, W. Zhu, and S. Yang, "Community preserving network embedding," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, 2017.

[72] Y. Liu, Q. Wang, X. Wang, F. Zhang, L. Geng, J. Wu, and Z. Xiao, "Community enhanced graph convolutional networks," *Pattern Recognition Letters*, vol. 138, pp. 462–468, 2020.

[73] M. Nickel, V. Tresp, and H.-P. Kriegel, "A three-way model for collective learning on multi-relational data," in *Icml*, 2011.

[74] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Neural Information Processing Systems (NIPS)*, 2013, pp. 1–9.

[75] B. Yang, W.-t. Yih, X. He, J. Gao, and L. Deng, "Embedding entities and relations for learning and inference in knowledge bases," *arXiv preprint arXiv:1412.6575*, 2014.

[76] D. Davidov, O. Tsur, and A. Rappoport, "Enhanced sentiment learning using twitter hashtags and smileys," in *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, ser. COLING '10. Association for Computational Linguistics, 2010, pp. 241–249.

[77] L. Hong, A. Ahmed, S. Gurumurthy, A. J. Smola, and K. Tsioutsiouliklis, "Discovering geographical topics in the twitter stream," in *Proceedings of the 21st International Conference on World Wide Web*, ser. WWW '12. ACM, 2012, pp. 769–778.

[78] H. Huang, Q. Zhang, X. Huang et al., "Mention recommendation for twitter with end-to-end memory network," in *Proc. IJCAI*, vol. 17, 2017, pp. 1872–1878.

[79] H. Alvari, "Twitter hashtag recommendation using matrix factorization," *arXiv preprint arXiv:1705.10453*, 2017.

[80] P. Veli ˇckovi ´c, G. Cucurull, A. Casanova, A. Romero, P. Li'o, and Y. Bengio, "Graph attention networks," in *Proceedings of ICLR 2018, the 6th International Conference on Learning Representations*, 2018.

[81] X. He, Z. He, J. Song, Z. Liu, Y.-G. Jiang, and T.-S. Chua, "Nais: Neural attentive item similarity model for recommendation," *IEEE Transactions on Knowledge and Data Engineering*, 2018.

[82] J. Chen, H. Zhang, X. He, L. Nie, W. Liu, and T.-S. Chua, "Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention," in *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 2017, pp. 335–344.

[83] E. Zangerle, W. Gassler, and G. Specht, "Recommending#-tags in twitter," in *Proceedings of the Workshop on Semantic Adaptive Social Web (SASWeb 2011). CEUR Workshop Proceedings*, vol. 730, 2011, pp. 67–78.

[84] Z. Ding, X. Qiu, Q. Zhang, and X. Huang, "Learning topical translation model for microblog hashtag suggestion." in *IJCAI*, 2013, pp. 2078–2084.

[85] Z. Ding, Q. Zhang, and X. Huang, "Automatic hashtag recommendation for microblogs using topic-specific translation model," *Proceedings of COLING 2012: Posters*, pp. 265–274, 2012.

[86] J. Weston, S. Chopra, and K. Adams, "# tagspace: Semantic embeddings from hashtags," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1822–1827.

[87] Y. Gong and Q. Zhang, "Hashtag recommendation using attention-based convolutional neural network." in *IJCAI*, 2016, pp. 2782–2788.

[88] N. Ben-Lhachemi and E. H. Nfaoui, "Using tweets embeddings for hashtag recommendation in twitter," *Procedia Computer Science*, vol. 127, pp. 7–15, 2018.

[89] K. Dey, R. Shrivastava, S. Kaushik, and L. V. Subramaniam, "Emtagger: A word embedding based novel method for hashtag recommendation on twitter," in *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, 2017, pp. 1025–1032.

[90] Q. Zhang, Y. Gong, X. Sun, and X. Huang, "Time-aware personalized hashtag recommendation on social media," in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, 2014, pp. 203–212.

[91] W. Feng and J. Wang, "We can learn your# hashtags: Connecting tweets to explicit topics," in *2014 IEEE 30th International Conference on Data Engineering*. IEEE, 2014, pp. 856–867.

[92] Y. Wei, Z. Cheng, X. Yu, Z. Zhao, L. Zhu, and L. Nie, "Personalized hashtag recommendation for micro-videos," *arXiv preprint arXiv:1908.09987*, 2019.

[93] Y. S. Rawat and M. S. Kankanhalli, "Contagnet: Exploiting user context for image tag recommendation," in *Proceedings of the 24th ACM international conference on Multimedia*. ACM, 2016, pp. 1102–1106.

[94] C. Chunseong Park, B. Kim, and G. Kim, "Attend to you: Personalized image captioning with context sequence memory networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 895–903.

[95] J. Tang, X. Hu, and H. Liu, "Social recommendation: a review," *Social Network Analysis and Mining*, vol. 3, no. 4, pp. 1113–1133, 2013.

[96] P. Symeonidis, E. Tiakas, and Y. Manolopoulos, "Product recommendation and rating prediction based on multi-modal social networks," in *Proceedings of the fifth ACM conference on Recommender systems*, 2011, pp. 61–68.

[97] J. Tang, H. Gao, X. Hu, and H. Liu, "Context-aware review helpfulness rating prediction," in *Proceedings of the 7th ACM Conference on Recommender Systems*, 2013, pp. 1–8.

[98] P. Massa and P. Avesani, "Trust-aware collaborative filtering for recommender systems," in *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*. Springer, 2004, pp. 492–508.

[99] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proceedings of the fourth ACM conference on Recommender systems*, 2010, pp. 135–142.

[100] J. Tang, H. Gao, and H. Liu, "mtrust: Discerning multi-faceted trust in a connected world," in *Proceedings of the fifth ACM international conference on Web search and data mining*, 2012, pp. 93–102.

[101] J. Tang, H. Gao, X. Hu, and H. Liu, "Exploiting homophily effect for trust prediction," in *Proceedings of the sixth ACM international conference on Web search and data mining*, 2013, pp. 53–62.

[102] H. Ma, H. Yang, M. R. Lyu, and I. King, "Sorec: social recommendation using probabilistic matrix factorization," in *Proceedings of the 17th ACM conference on Information and knowledge management*, 2008, pp. 931–940.

[103] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *international conference on learning representations*, 2015.

[104] T. Rocktäschel, E. Grefenstette, K. M. Hermann, T. K. iský, and P. Blunsom, "Reasoning about entailment with neural attention," *international conference on learning representations*, 2016.

[105] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo, "Image captioning with semantic attention," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4651–4659.

[106] A. M. Rush, S. Chopra, and J. Weston, "A neural attention model for abstractive sentence summarization," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 379–389.

[107] J. Lu, C. Xiong, D. Parikh, and R. Socher, "Knowing when to look: Adaptive attention via a visual sentinel for image captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 6, 2017, p. 2.

[108] T. Qiao, J. Dong, and D. Xu, "Exploring human-like attention supervision in visual question answering," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[109] A. Das, H. Agrawal, L. Zitnick, D. Parikh, and D. Batra, "Human attention in visual question answering: Do humans and deep networks look at the same regions?" *Computer Vision and Image Understanding*, vol. 163, pp. 90–100, 2017.

[110] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowledge-based systems*, vol. 46, pp. 109–132, 2013.

[111] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proceedings of the 26th International Conference on World Wide Web*, ser. WWW '17. International World Wide Web Conferences Steering Committee, 2017, pp. 173–182.

[112] L. Liu, M. Utiyama, A. M. Finch, and E. Sumita, "Neural machine translation with supervised attention." *international conference on computational linguistics*, pp. 3093–3102, 2016.

[113] Y. Zhang, Y. Xiong, X. Kong, S. Li, J. Mi, and Y. Zhu, "Deep collective classification in heterogeneous information networks," in *Proceedings of the 2018 World Wide Web Conference*, ser. WWW '18. International World Wide Web Conferences Steering Committee, 2018, pp. 399–408.

[114] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[115] H. Huang, Q. Zhang, Y. Gong, and X. Huang, "Hashtag recommendation using end-to-end memory networks with hierarchical attention," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 2016.

[116] X. He, T. Chen, M.-Y. Kan, and X. Chen, "Trirank: Review-aware explainable recommendation by modeling aspects," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, ser. CIKM '15. ACM, 2015, pp. 1661–1670.

[117] H.-M. Chen, M.-H. Chang, P.-C. Chang, M.-C. Tien, W. H. Hsu, and J.-L. Wu, "Sheepdog: group and tag recommendation for flickr photos by automatic search-based learning," in *ACM Multimedia*, 2008.

[118] R. Krestel, P. Fankhauser, and W. Nejdl, "Latent dirichlet allocation for tag recommendation," in *Proceedings of the Third ACM Conference on Recommender Systems*, ser. RecSys '09. ACM, 2009, pp. 61–68.

[119] Y. Gong and Q. Zhang, "Hashtag recommendation using attention-based convolutional neural network," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, ser. IJCAI'16. AAAI Press, 2016, pp. 2782–2788.

[120] Z. Wang, J. Liao, Q. Cao, H. Qi, and Z. Wang, "Friendbook: a semantic-based friend recommendation system for social networks," *IEEE transactions on mobile computing*, vol. 14, no. 3, pp. 538–551, 2014.

[121] G. Iyer, D. Soberman, and J. M. Villas-Boas, "The targeting of advertising," *Marketing Science*, vol. 24, no. 3, pp. 461–476, 2005.

[122] D. Agarwal, B.-C. Chen, Q. He, Z. Hua, G. Lebanon, Y. Ma, P. Shivaswamy, H.-P. Tseng, J. Yang, and L. Zhang, "Personalizing linkedin feed," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, pp. 1651–1660.

[123] N. Morris, A. Basso, T. Hines, A. Ivanov, and J. Mikan, "Method and apparatus for targeted advertising," Dec. 27 2012, uS Patent App. 13/167,998.

[124] A. Pak and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining." in *LREc*, vol. 10, no. 2010, 2010.

[125] C. D. Manning, H. Schütze, and P. Raghavan, *Introduction to information retrieval*. Cambridge university press, 2008.

[126] J. Dalton, L. Dietz, and J. Allan, "Entity query feature expansion using knowledge base links," in *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, 2014, pp. 365–374.

[127] S. Settle, K. Levin, H. Kamper, and K. Livescu, "Query-by-example search with discriminative neural acoustic word embeddings," *arXiv preprint arXiv:1706.03818*, 2017.

[128] N. Borjian, "Query-by-example music information retrieval by score-based genre prediction and similarity measure," *International Journal of Multimedia Information Retrieval*, vol. 6, no. 2, pp. 155–166, 2017.

[129] G. Piao and J. G. Breslin, "Leveraging followee list memberships for inferring user interests for passive users on twitter," in *Proceedings of the 28th ACM Conference on Hypertext and Social Media*. ACM, 2017, pp. 155–164.

[130] Z. Zhang, P. Cui, and W. Zhu, "Deep learning on graphs: A survey," *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[131] A. Onan, S. Korukoglu, and H. Bulut, "Lda-based topic modelling in text sentiment classification: An empirical analysis." *Int. J. Comput. Linguistics Appl.*, vol. 7, no. 1, pp. 101–119, 2016.

[132] Y. Cha, B. Bi, C.-C. Hsieh, and J. Cho, "Incorporating popularity in topic models for social network analysis," in *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2013, pp. 223–232.

[133] K. Henderson and T. Eliassi-Rad, "Applying latent dirichlet allocation to group discovery in large graphs," in *Proceedings of the 2009 ACM symposium on Applied Computing*, 2009, pp. 1456–1461.

[134] H. Deng, J. Han, B. Zhao, Y. Yu, and C. X. Lin, "Probabilistic topic models with biased propagation on heterogeneous information networks," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2011, pp. 1271–1279.

[135] S. Li and H. Zhao, "A survey on representation learning for user modeling," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, 2020, pp. 4997–5003.

[136] S. Kanoje, S. Girase, and D. Mukhopadhyay, "User profiling trends, techniques and applications," *arXiv preprint arXiv:1503.07474*, 2015.

[137] S. Liang, X. Zhang, Z. Ren, and E. Kanoulas, "Dynamic embeddings for user profiling in twitter," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 1764–1773.

[138] Y. Cha and J. Cho, "Social-network analysis using topic models," in *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2012, pp. 565–574.

[139] P. Bhattacharya, M. B. Zafar, N. Ganguly, S. Ghosh, and K. P. Gummadi, "Inferring user interests in the twitter social network," in *Proceedings of the 8th ACM Conference on Recommender systems*. ACM, 2014, pp. 357–360.

[140] C. Besel, J. Schlötterer, and M. Granitzer, "Inferring semantic interest profiles from twitter followees: does twitter know better than your friends?" in *Proceedings of the 31st Annual ACM symposium on applied computing*. ACM, 2016, pp. 1152–1157.

[141] J. J. Whang, D. F. Gleich, and I. S. Dhillon, "Overlapping community detection using seed set expansion," in *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, 2013, pp. 2099–2108.

[142] M. Hamann, E. Röhrs, and D. Wagner, "Local community detection based on small cliques," *Algorithms*, vol. 10, no. 3, p. 90, 2017.

[143] C.-S. Chang, C.-J. Chang, W.-T. Hsieh, D.-S. Lee, L.-H. Liou, and W. Liao, "Relative centrality and local community detection," *Network Science*, vol. 3, no. 4, pp. 445–479, 2015.

[144] D. Mottin and E. Müller, "Graph exploration: From users to large graphs," in *proceedings of the 2017 ACM international conference on management of data*, 2017, pp. 1737–1740.

[145] X. Yi and J. Allan, "Evaluating topic models for information retrieval," in *Proceedings of the 17th ACM conference on Information and knowledge management*, 2008, pp. 1431–1432.

[146] F. Jian, J. X. Huang, J. Zhao, T. He, and P. Hu, "A simple enhancement for ad-hoc information retrieval via topic modelling," in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, 2016, pp. 733–736.

[147] X. Yi and J. Allan, "A comparative study of utilizing topic models for information retrieval," in *European conference on information retrieval*. Springer, 2009, pp. 29–41.

[148] G. Cao, J.-Y. Nie, J. Gao, and S. Robertson, "Selecting good expansion terms for pseudo-relevance feedback," in *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, 2008, pp. 243–250.

[149] F. D. Malliaros and M. Vazirgiannis, "Clustering and community detection in directed networks: A survey," *Physics reports*, vol. 533, no. 4, pp. 95–142, 2013.

[150] H. Yin, A. R. Benson, J. Leskovec, and D. F. Gleich, "Local higher-order graph clustering," in *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, 2017, pp. 555–564.

[151] J. Yang and J. Leskovec, "Patterns of temporal variation in online media," in *Proceedings of the fourth ACM international conference on Web search and data mining*, 2011, pp. 177–186.

[152] J. Song, T. He, L. Gao, X. Xu, A. Hanjalic, and H. T. Shen, "Binary generative adversarial networks for image retrieval," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[153] W. Zhang, "Generative adversarial nets for information retrieval: Fundamentals and advances," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 2018, pp. 1375–1378.

[154] X. Yang, D. Lo, X. Xia, L. Bao, and J. Sun, "Combining word embedding with information retrieval to recommend similar bug reports," in *2016 IEEE 27Th international symposium on software reliability engineering (ISSRE)*. IEEE, 2016, pp. 127–137.

[155] X. Ye, H. Shen, X. Ma, R. Bunescu, and C. Liu, "From word embeddings to document similarities for improved information retrieval in software engineering," in *Proceedings of the 38th international conference on software engineering*, 2016, pp. 404–415.

[156] Z. Kalinic and V. Marinkovic, "Determinants of users' intention to adopt m-commerce: an empirical analysis," *Information Systems and e-Business Management*, vol. 14, no. 2, pp. 367–387, 2016.

[157] P. Wang, B. Xu, Y. Wu, and X. Zhou, "Link prediction in social networks: the state-of-the-art," *Science China Information Sciences*, vol. 58, no. 1, pp. 1–38, 2015.

[158] P. Agrawal, V. K. Garg, and R. Narayanam, "Link label prediction in signed social networks." in *IJCAI*, 2013.

[159] P. Massa and P. Avesani, "Trust-aware recommender systems," in *Proceedings of the 2007 ACM conference on Recommender systems*, 2007, pp. 17–24.

[160] S. Kumar, F. Spezzano, and V. Subrahmanian, "Accurately detecting trolls in slashdot zoo via decluttering," in *Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on*. IEEE, 2014, pp. 188–195.

[161] C.-J. Hsieh, K.-Y. Chiang, and I. S. Dhillon, "Low rank modeling of signed networks," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012, pp. 507–515.

[162] D. Cartwright and F. Harary, "Structural balance: a generalization of heider's theory." *Psychological review*, vol. 63, no. 5, p. 277, 1956.

[163] K.-Y. Chiang, N. Natarajan, A. Tewari, and I. S. Dhillon, "Exploiting longer cycles for link prediction in signed networks," in *Proceedings of the 20th ACM international conference on Information and knowledge management*. ACM, 2011, pp. 1157–1162.

[164] M. R. Islam, B. A. Prakash, and N. Ramakrishnan, "Signet: Scalable embeddings for signed networks," *arXiv preprint arXiv:1702.06819*, 2017.

[165] J. G. Booth, G. Casella, and J. P. Hobert, "Clustering using objective functions and stochastic search," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 70, no. 1, pp. 119–139, 2008.

[166] B. Croft and J. Lafferty, *Language modeling for information retrieval*. Springer Science & Business Media, 2013, vol. 13.

[167] X. Liu and W. B. Croft, "Cluster-based retrieval using language models," in *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2004, pp. 186–193.

[168] D. Guthrie, B. Allison, W. Liu, L. Guthrie, and Y. Wilks, "A closer look at skip-gram modelling," in *Proceedings of the 5th international Conference on Language Resources and Evaluation (LREC-2006)*, 2006, pp. 1–4.

[169] C. Zhai and J. Lafferty, "A study of smoothing methods for language models applied to information retrieval," *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 2, pp. 179–214, 2004.

[170] T. Chen, H. Yin, H. Chen, R. Yan, Q. V. H. Nguyen, and X. Li, "Air: Attentional intention-aware recommender systems," in *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE, 2019, pp. 304–315.

[171] P. Cui, X. Wang, J. Pei, and W. Zhu, "A survey on network embedding," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 5, pp. 833–852, 2018.

[172] F. Heider, "Attitudes and cognitive organization," *The Journal of psychology*, vol. 21, no. 1, pp. 107–112, 1946.

[173] D. Cartwright and F. Harary, "Structural balance: a generalization of heider's theory." *Psychological review*, vol. 63, no. 5, p. 277, 1956.

[174] J. Leskovec, D. Huttenlocher, and J. Kleinberg, "Signed networks in social media," in *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 2010, pp. 1361–1370.

[175] H. Wang, F. Zhang, M. Hou, X. Xie, M. Guo, and Q. Liu, "Shine: Signed heterogeneous information network embedding for sentiment link prediction," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. ACM, 2018, pp. 592–600.

[176] F. Xue, X. He, X. Wang, J. Xu, K. Liu, and R. Hong, "Deep item-based collaborative filtering for top-n recommendation," *ACM Transactions on Information Systems (TOIS)*, vol. 37, no. 3, p. 33, 2019.

[177] M. Ou, P. Cui, J. Pei, Z. Zhang, and W. Zhu, "Asymmetric transitivity preserving graph embedding," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2016, pp. 1105–1114.

[178] M. R. Islam, B. A. Prakash, and N. Ramakrishnan, "Signet: Scalable embeddings for signed networks," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2018, pp. 157–169.

[179] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *Proceedings of the 24th international conference on world wide web*. International World Wide Web Conferences Steering Committee, 2015, pp. 1067–1077.

[180] T. Derr, C. Wang, S. Wang, and J. Tang, "Relevance measurements in online signed social networks," in *In Proceedings of the 14th International Workshop on Mining and Learning with Graphs (MLG)*, 2018.

[181] J. Jung, W. Jin, L. Sael, and U. Kang, "Personalized ranking in signed networks using signed random walk with restart," in *2016 IEEE 16th International Conference on Data Mining (ICDM)*. IEEE, 2016, pp. 973–978.

[182] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Computing Surveys (CSUR)*, vol. 52, no. 1, p. 5, 2019.

[183] V. Mnih, N. Heess, A. Graves et al., "Recurrent models of visual attention," in *Advances in neural information processing systems*, 2014, pp. 2204–2212.

[184] M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi, "Bidirectional attention flow for machine comprehension," *arXiv preprint arXiv:1611.01603*, 2016.

[185] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[186] J. Huang, H. Shen, L. Hou, and X. Cheng, "Signed graph attention networks," *arXiv preprint arXiv:1906.10958*, 2019.

[187] J. Zhang, X. Shi, J. Xie, H. Ma, I. King, and D.-Y. Yeung, "Gaan: Gated attention networks for learning on large and spatiotemporal graphs," *arXiv preprint arXiv:1803.07294*, 2018.

[188] J. B. Lee, R. A. Rossi, S. Kim, N. K. Ahmed, and E. Koh, "Attention models in graphs: A survey," *arXiv preprint arXiv:1807.07984*, 2018.

[189] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Advances in Neural Information Processing Systems*, 2016, pp. 3844–3852.

[190] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, "Convolutional networks on graphs for learning molecular fingerprints," in *Advances in neural information processing systems*, 2015, pp. 2224–2232.

[191] J. Bruna, W. Zaremba, A. Szlam, and Y. Lecun, "Spectral networks and locally connected networks on graphs," in *International Conference on Learning Representations (ICLR), CBLS, April 2014*, 2014.

[192] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016.

[193] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Advances in Neural Information Processing Systems*, 2017, pp. 1025–1035.

[194] Q. Wang, Z. Mao, B. Wang, and L. Guo, "Knowledge graph embedding: A survey of approaches and applications," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 12, pp. 2724–2743, 2017.

[195] B. Shams and S. Haratizadeh, "Sibrank: Signed bipartite network analysis for neighbor-based collaborative ranking," *Physica A: Statistical Mechanics and its Applications*, vol. 458, pp. 364–377, 2016.

[196] T. Derr, C. Johnson, Y. Chang, and J. Tang, "Balance in signed bipartite networks," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 1221–1230.

[197] S. Chang, W. Han, J. Tang, G.-J. Qi, C. C. Aggarwal, and T. S. Huang, "Heterogeneous network embedding via deep architectures," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2015, pp. 119–128.

[198] P. Esmailian and M. Jalili, "Community detection in signed networks: the role of negative ties in different scales," *Scientific reports*, vol. 5, p. 14339, 2015.

[199] C. Liu, J. Liu, and Z. Jiang, "A multiobjective evolutionary algorithm based on similarity for community detection from signed social networks," *IEEE transactions on cybernetics*, vol. 44, no. 12, pp. 2274–2287, 2014.

[200] M. Shahriari and M. Jalili, "Ranking nodes in signed social networks," *Social network analysis and mining*, vol. 4, no. 1, p. 172, 2014.

[201] Z. Wu, C. C. Aggarwal, and J. Sun, "The troll-trust model for ranking in signed networks," in *Proceedings of the Ninth ACM international conference on Web Search and Data Mining*. ACM, 2016, pp. 447–456.

[202] X. Yang, H. Steck, and Y. Liu, "Circle-based recommendation in online social networks," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2012, pp. 1267–1275.

[203] S. Gauch, M. Speretta, A. Chandramouli, and A. Micarelli, "User profiles for personalized information access," *The adaptive web*, pp. 54–89, 2007.

[204] X. Shen, B. Tan, and C. Zhai, "Implicit user modeling for personalized search," in *Proceedings of the 14th ACM international conference on Information and knowledge management*, 2005, pp. 824–831.

[205] H. Liu, Z. Hu, A. Mian, H. Tian, and X. Zhu, "A new user similarity model to improve the accuracy of collaborative filtering," *Knowledge-Based Systems*, vol. 56, pp. 156–166, 2014.

[206] Y. Zhang and X. Chen, "Explainable recommendation: A survey and new perspectives," *arXiv preprint arXiv:1804.11192*, 2018.

[207] S. Ganguly and V. Pudi, "Paper2vec: Combining graph and text information for scientific paper representation," in *European conference on information retrieval*. Springer, 2017, pp. 383–395.

[208] X. Han, Z. Liu, and M. Sun, "Neural knowledge acquisition via mutual attention between knowledge graph and text," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.