공학박사 학위논문

Mining Latent Networks on Social Network and E-Commerce Platforms

소셜 네트워크와 이커머스 플랫폼에서의 잠재 네트워크 마이닝

2023 년 2 월

서울대학교 대학원

전기 · 컴퓨터공학부

변 형 호

Mining Latent Networks on Social Network and E-Commerce Platforms

소셜 네트워크와 이커머스 플랫폼에서의 잠재 네트워크 마이닝

지도교수 권 태 경

이 논문을 공학박사 학위논문으로 제출함

2023 년 1 월

서울대학교 대학원

전기 · 컴퓨터공학부

변 형 호

변형호의 공학박사 학위논문을 인준함

2022 년 12 월

위 원 장 이 상 구 (인)

부위원장 권 태 경 (인)

위 원 김 종 권 (인)

위 원 강 유 (인)

위 원 이 규 행 (인)

Abstract

Mining Latent Networks on Social Network and E-Commerce Platforms

Hyungho Byun

Department of Computer Science & Engineering

The Graduate School

Seoul National University

Following the exploding usage on online services, people are connected with each other more broadly and widely. In online platforms, people influence each other, and have tendency to reflect their opinions in decision-making. Social Network Services (SNSs) and E-commerce are typical example of online platforms.

User behaviors in online platforms can be defined as relation between user and platform components. A user's purchase is a relationship between a user and a product, and a user's check-in is a relationship between a user and a place. Here, information such as action time, rating, tag, etc. may be included. In many studies, platform user behavior is represented in graph form. At this time, the elements constituting the nodes of the graph are composed of objects such as users and products and places within the platform, and the interaction between the platform elements and the user can be expressed as two nodes being connected.

In this study, I present studies to identify potential networks that affect the user's behavior graph defined on the two platforms.

In ANES, I focus on representation learning for social link inference based on user trajectory data. While traditional methods predict relations between users by considering hand-crafted features, recent studies first perform representation learning using network/node embedding or graph neural networks (GNNs) for downstream tasks such as node classification and link prediction. However, those approaches fail to capture behavioral patterns of individuals ingrained in periodical visits or long-distance movements. To better learn behavioral patterns, this paper proposes a novel scheme called ANES (Aspect-oriented Network Embedding for Social link inference). ANES learns aspect-oriented relations between users and Point-of-Interests (POIs) within their contexts. ANES is the first approach that extracts the complex behavioral pattern of users from both trajectory data and the structure of User-POI bipartite graphs. Extensive experiments on several real-world datasets show that ANES outperforms state-of-the-art baselines.

In contrast to active social networks, people are connected to other users regardless of their intentions in some platforms, such as online shopping websites and restaurant review sites. They do not have any information about each other in advance, and they only have a common point which is that they have visited or have planned to visit same place or purchase a product. Interestingly, users have tendency to be influenced by the review data on their purchase intentions.

Unfortunately, this instinct is easily exploited by opinion spammers. In SC-Com, I focus on opinion spam detection in online shopping services. In many

cases, my decision-making process is closely related to online reviews. However, there have been threats of opinion spams by hired reviewers increasingly, which aim to mislead potential customers by hiding genuine consumers' opinions. Opinion spams should be filed up collectively to falsify true information. Fortunately, I propose the way to spot the possibility to detect them from their collusiveness. In this paper, I propose SC-Com, an optimized collusive community detection framework. It constructs the graph of reviewers from the collusiveness of behavior and divides a graph by communities based on their mutual suspiciousness. After that, I extract community-based and temporal abnormality features which are critical to discriminate spammers from other genuine users. I show that my method detects collusive opinion spam reviewers effectively and precisely from their collective behavioral patterns. In the real-world dataset, my approach showed prominent performance while only considering primary data such as time and ratings.

These implicit network inference models studied on various data in this thesis predicts users who are likely to be pre-connected to unlabeled data, so it is expected to contribute to areas such as advertising recommendation systems and malicious user detection by providing useful information.

Keywords: Social network analysis, Spam detection, Graph learning, Location-based social networks, Social link inference

Student Number: 2016-21210

Contents

Abstract · i

Contents · iv

List of Tables · vi

List of Figures · vii

Chapter 1 Introduction · · · · · · · · · · · · · · · · · · 1

Chapter 2 Social link Inference in Location−based check−in data · 5

2.1 Background · 5

2.2 Related Work · 12

2.3 Location−based Social Network Service Data · · · 15

2.4 Aspect−wise Graph Decomposition · · · · · · · · 18

2.5 Aspect−wise Graph learning · · · · · · · · · · · 19

2.6 Inferring Social Relation from User Representation 21

2.7 Performance Analysis · · · · · · · · · · · · · · · 23

2.8 Discussion and Implications · · · · · · · · · · · · 26

2.9 Summary · 34

Chapter 3 Detecting collusiveness from reviews in Online platforms and its application · · · · · · · · · · · · · · · · 35

3.1 Background · 35

3.2 Related Work · 39

3.3 Online Review Data · · · · · · · · · · · · · · · · · · 43

3.4 Collusive Graph Projection · · · · · · · · · · · · · · 44

3.5 Reviewer Community Detection · · · · · · · · · · · 47

3.6 Review Community feature extraction and spammer detection · 51

3.7 Performance Analysis · · · · · · · · · · · · · · · · 53

3.8 Discussion and Implications · · · · · · · · · · · · · 55

3.9 Summary · 62

Chapter 4 Conclusion · 63

Bibliography

초 록

List of Tables

Table 2.1 Statistics of dataset. Gowalla and Brightkite do not contain category data.

Table 2.2 Performance of ANES with different subsets aspects. T, C, and G denote time slot, categorical data, and geographical data, respectively.

Table 2.3 Top similar/dissimilar pairs of timeslots

Table 3.1 Values of optimized parameters in SC-Com

Table 3.2 Features extracted from detected communities. Shared features mean all members in a community have the same value, while Individual features are different individually. My original features are noted as *.

Table 3.3 Table 3.3. Features extracted from user's review data. My original features are noted as *.

Table 3.4 Summary of baselines with their methods

Table 3.5 Classification performance by varying the scope of features

Table 3.6 Classification performance with baselines

Table 3.7 Classification Performance of SC-Com in Camouflaging Scenario

Table 3.8 Classification Performance of SC-Com in Compromised Accounts Scenario

List of Figures

Figure 2.1 Co-visitation distribution of friends in different time slots. In Gowalla-Austin, more than 60 percent of social links do not have a co-visitation within three days. In Brightkite-NYC, the ratio of social links which have never had co-visitation is about 86%.

Figure 2.2 Architecture of ANES. Multiple aspects of visitation are decomposed into different bipartite graphs. Each graph is embedded by a context-aware embedding scheme.

Figure 2.3 Performance comparison of ANES in 8 LBSN dataset.

Figure 2.4 Performance of social relation inference at different co-visitation intervals. ANES effectively infers social relation in both situations when there is no co-visitation between user pairs and when the co-visitation interval is within 2-hours.

Figure 2.5 Performance on different proportion of data on three datasets

Figure 2.6 Visualization of time slot embedding using t-SNE. The upper figure is the projected position of each relation embedding in 3-D space, and the lower figure is an enlarged version of each embedding to show which relations are close to each other.

Figure 3.1 SC-Com utilizes community-based data from user similarity projection graph and review data to classify reviews as spams and benign reviews.

Figure 3.2 The result of community detection. In every λ, the overall purity of communities is higher than 0.9. It means detected communities are mostly spammer groups, or benign groups. Second figure depicts the number of spammers and benign users as λ varies.

Figure 3.3 Cumulative distribution of community-based feature in all reviewers when λ=1.7.

Figure 3.4 Cumulative Distribution of Individual Behavior-based Features

Figure 3.5 Parameter Sensitivity of λ. As my framework shows the best performance when λ=1.7 with the highest F1, the overall performance remains stable when λ varies

Chapter 1 Introduction

Following the exploding usage on online services, people are connected with each other more broadly and widely. In online platforms, people influence each other, and have tendency to reflect their opinions in decision-making. That is the reason why many researches has been proposed in a category of social recommendation [57]. Social Network Services (SNSs) and E-commerce are typical example of online platforms. Analyzing user behaviors on those platforms has gained expansive research attention. By utilizing social relationships, I can predict person's behaviors [50], recommend purchases of products [38, 47], and provide customized advertising services [43]. Those applications furnish new marketing and service opportunities to both service providers and users.

User behaviors in online platforms can be defined as relation between user and platform components. A user's purchase is a relationship between a user and a product, and a user's check-in is a relationship between a user and a place. Here, information such as action time, rating, tag, etc. may be included. In many studies, platform user behavior is represented in graph form. At this time, the elements constituting the nodes of the graph are composed of objects such as users and products and places within the platform, and the interaction between the platform elements and the user can be expressed as two nodes being connected.

In this study, I present studies to identify potential networks that affect the user's behavior graph defined on the two platforms. Before introducing each study, I explain the different characteristics of two platforms, by focusing on the connection of each platform. The connection between users in two platforms can be regarded as two types according to the subject of the relationship: the active social networks and passive social networks.

If users determine who to connect, it is an active social network. Following/follower relation and subscription can be examples of active social network. In active social network, users relate by their own decision. This relation can be found in general SNSs. In this paper, I only consider reciprocal link in SNS, to focus on relation with influence both sides.

In active social networks, the main principle of why people connect is homophily. According to [58], relation between people with similar characteristics occur more often than people with dissimilar characteristics. Those insight become a theoretical underlying of graph neural networks (GNNs). In this paper, I consider location-based social networks (LBSNs) as examples of active social networks and find the way to infer social relation between users. In the case of a social network service, many posts are created in the form of a check-in visit to a specific place, and a user's visit to a place is greatly influenced by a pre-existing friend relationship between users. Identifying the relationship between users latent in the user activity network can be helpful in predicting activity.

Previous approaches [12,5] utilized homophily as the insight that users are likely to be connected on LBSNs when they actually meet in real life. It is called co-visitation. However, I proved that homophily can be applied to the similarity of behavioral patterns, which is beyond simple co-visitation. Even though users are not actually connected simultaneously, I hypothesized that similar behavioral patterns are likely to have a social link.

I propose a novel method called ANES (Aspect-oriented Network Embedding for Social link inference), a representation learning method customized for unsupervised learning scenarios. First, I regard trajectory data as a directed bipartite graph of users and POIs. Users and POIs are connected if there is a trajectory record between them. Features of an edge between users and POIs are defined by the aspect of each trajectory from additional information such as time slots, geographic information, and activity categories. I decompose the graph into aspect-wise bipartite subgraphs, each of which is learned separately by a context-aware embedding scheme.

One main point of the context-aware embedding scheme is that if a user visits a POI in a certain context, they should be close to each other in the embedding space by having a similarly transformed vector from their context. Different from previous approaches, which only focused on co-visitations of users, I learn the behavioral characteristics of users by considering contexts between users and POIs. First, user embedding is projected by each context to model the user's behavioral representation. After that, I define a transition-based relation between the projected user and the POI. In the context-aware embedding scheme, relations between users are modeled as a mixed form of projection and transition. Users' complex behavioral patterns in various contexts can be learned in this method. Context-aware embedding can lead to accurate representation learning useful for inferring hidden social relations between users without explicit label information in advance. After that, I measure the probability of social relations by comparing a pair of user embedding by concatenating the aspect-wise user embedding together. ANES works well in the unsupervised condition by effectively defining loss function from the relation between users and POIs, to model complex behavioral patterns of users upon various aspects. The detailed explanation of this work is in Chapter 2.

In contrast to active social networks, people are connected to other users regardless of their intentions in e-commerce platforms, such as online shopping websites and restaurant review sites. They do not have any information about each other in advance, and they only have a common point which is that they have visited or have planned to visit same place or purchase a product. Interestingly, users have tendency to be influenced by the review data on their purchase intentions. However, this instinct of human has been exploited by malicious reviews. Some service operators provide users with material rewards such as free products, deliberately soliciting high-rating reviews, or mobilizing accounts rather than actual users to reduce the ratings of certain products. This act of calling review spamming is now very prevalent. By piling up overwhelmingly positive fake reviews, they induce more customers to visit.

Paradoxically, spamming is generally based on collusion. To achieve the boost of the target rating in a short time, or to undermine the reputation of the competitor, multiple accounts must pile up reviews with the same polarity in a short time. To solve the spam detection problem, I focused on the above-mentioned public offering, identified the public offering relationship between users on the online shopping platform, and used it in research to find accounts used for actual spamming.

I propose SC-Com, a robust spam detection method which can be used in many domains by only considering primary data such as time data and ratings. SC-Com has caught opinion spam reviewers effectively and precisely in large-scale real-world data. It aims to extract communities of reviewers with their behavioral collusiveness and utilizes features from each community to supervised learning for detecting opinion spammers. First, I observe the importance of finding collusiveness in opinion spam, by constructing a projection graph of reviewers according to their behavioral similarities. Second, I propose the framework to divide whole users into closely related communities and discriminate each community by extracting features related to communities' suspiciousness. Lastly, I prove that my approach can provide a reliable and robust solution with a high F1 score of 0.93 in a real-world dataset.

Chapter 2 describes ANES, an unsupervised learning-based approach that predicts connectivity between users in location-based social networks Chapter 3 proposes SC-Com, a method of separating the entire user into a community of users with similar behaviors by utilizing only the score and writing time of the review on an online shopping platform, and using the information obtained here for spam detection. Finally, Chapter 4 concludes the studies covered in this paper, and discusses implications and future development directions.

Chapter 2 Social Link Inference in Location-based Check-in Data

2.1 Background

I have witnessed explosive and wide deployments of a plethora of online social network services (SNSs) such as Twitter, Facebook, Instagram, and

Foursquare during the last two decades. These days, people can interact with each other without meeting in person through online social networks, and the range of human behavior and interactions are broadened. Unlike traditional face-to-face social relations, I enjoy the opportunity to observe and collect human interactions on SNS explicitly and implicitly.
Users explicitly specify relations such as following/subscription, which provide critical information to identify the social network. As the social relation data are accumulated, the problem of inferring social interactions has gained expansive research attention. By utilizing social relationships, I can predict a person's behavior [50], recommend purchases of products [38, 47], and provide customized advertising services [43]. Those applications come up with insights and opportunities for service providers and users.

As various location-based social networks (LBSNs) and smart devices have been widely adopted, trajectory data have been collected and they have attracted a great attention due to the powerful possibility of a variety of applications. For example, POI recommendation [49, 50, 51, 2] or next visit prediction [52] has been actively studied. Also, mining social links from trajectory data is of great value in both academic and industrial domains, e.g., graph completion tasks [55], personalized social recommendations [10], and advertisements [4].

Traditional studies on mining social links from trajectory data have focused on predicting pairwise relations without considering network structures [34, 44]. They rely on the co-visitations history of users to infer social links between users. These approaches are rooted in the common belief that friends tend to get along together and share their time in the same place [12, 5]. One critical problem of these schemes is the cold-start problem; a problem incurred by data insufficiency. As check-in data in LBSNs are created by voluntary participation and posting of users, data is inherently selective and partial. Furthermore, co-visitation-based approaches adopt arbitrary definitions of co-visitation, ranging from a simultaneous visit to a co-visitation within three days [29, 45]. Inconsistency in definitions can be regarded as a lack of generality when they are applied in other domains in the real world.

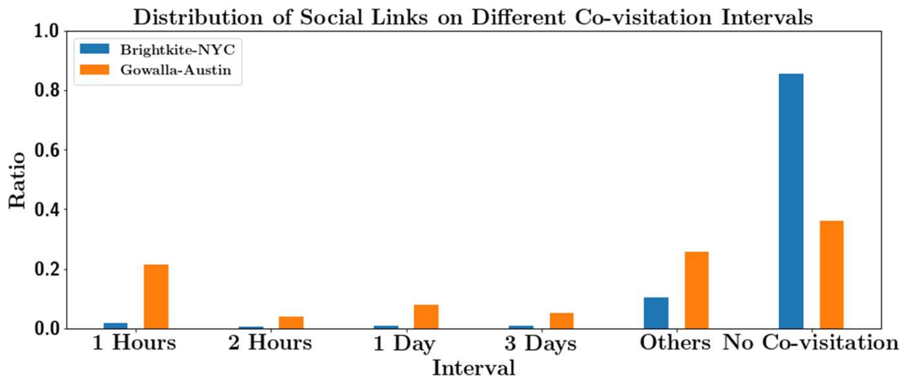


Figure 2.1. Co-visitation distribution of friends in different time slots. In Gowalla-Austin, more than 60 percent of social links do not have a co-visitation within three days. In Brightkite-NYC, the ratio of social links which have never had co-visitation is about 86%.

Lastly, co-visitation- based approaches consider opportunistic encountering of unknown persons as a genuine social interaction, especially when the social activity of people decreases due to situations like social distancing. I show this threat of misunderstanding from the data of LBSNs. I can observe that co-visitation cannot fully cover the friendship between users in the

Gowalla dataset, which is one of the most popular LBSN applications. In Figure 2.1, I depict the distribution of social links based on different co-visitation intervals. Even though I increase time intervals for defining co-visitations up to three days, a notable amount of friends do not visit the same POIs during the time interval. In the Gowalla-Austin dataset, which I use in the experiment for this research, the ratio of real friends who never makes co-visitations is 36%. Furthermore, in the Brightkite-NYC dataset, this ratio increases to 86%. This evidence strongly implies that co-visitation-oriented techniques may fail to fully understand the characteristics of social relations in LBSNs. This phenomenon necessitates to inferring implicit relationships to boost the performance of social link prediction on LBSNs. If I consider the social link inference task as a supervised link prediction task, inferring implicit relationships may be handy. Heuristics such as common neighbors or preferential attachment [3] have been widely used in link prediction. However, I aim to consider this problem as unsupervised learning for wide and generic applicability in various domains without knowing their social relations.

Recent studies on inferring users' implicit relations can be classified into two classes; graph neural networks (GNNs) and network embedding-based methods. Both methods start by representing LBSN data as graphs such as user-POI visitation graphs and user-user co-visitation graphs. Using GNN, some models consider the implicit relation between nodes in the network structure by utilizing aggregation and message-passing schemes [25, 15]. The theoretical underlying of GNNs is homophily, direct neighbors share similar characteristics, and they aim to make the embedding of nodes to be similar to their neighbors. Many network embedding methods [1, 49, 45] which are inspired by the success of Deepwalk [33], Node2Vec[18] and LINE

[41] extract the information embedded in the graph structure while preserving the node information effectively. However, several points restrict the generalization and applicability of those methods. First, a part of social relations should be visible to learn embeddings [49], or they require complicated pre-training for graph convolution network [45]. In addition, [1] does not model the complex dynamics of human visitation behavior. Especially the essential need for labeled social relation is a critical limitation in real-world data because labeled data may be protected due to privacy issues in many domains or even may not exists. Even though explicit labels are scarce, there might be latent relations between users that are not visible [31]. In that case, I can infer latent networks from users' trajectory data, without the ground truth of social relationships. A recent result [45] is a seminal method that considers unsupervised social link prediction in LBSN graphs. The authors proposed a graph convolution-based method using R-GCN[36] in the co-occurrence graph of users and POIs. It partitions the trajectory of users into a series of sub-trajectories and utilizes the pretrained result as an input of the graph convolution network.

However, they consider only the similarity and temporal distance of users' visitation ignoring to extract behavioral patterns according to contexts of each visitation. Suppose that there is a user who lives in a suburb and works downtown on weekdays. She is likely to spend time downtown during the day on weekdays, but she tends to stay near her home during the nights and weekends. If I apply the traditional network embedding approaches, her embedding should be close to both downtown and the suburb, so that those three embedding vectors can be close to each other. This in turn results in information ambiguity that cannot separate her visitation and workplace.

Fortunately, if I consider multiple contexts of user embedding separately, I can effectively satisfy both conditions. Note that the context of each visitation is not related to a single aspect only. User behaviors are related to many aspects such as purposes of visitations and their locations. Contrary to prior schemes that focus on a single aspect, a scheme that comprehend multiple aspects in a harmonious way may enhance the prediction accuracy.

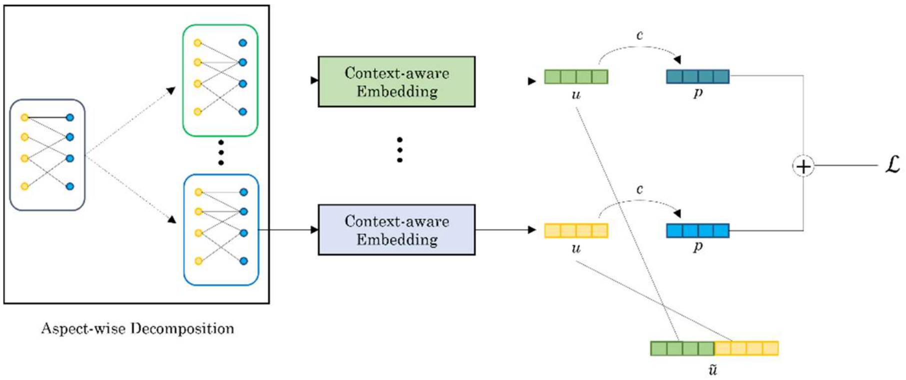


Figure 2.2. Architecture of ANES. Multiple aspects of visitation are decomposed into different bipartite graphs. Each graph is embedded by a context-aware embedding scheme.

In this paper, I propose a novel method called ANES (Aspect-oriented Network Embedding for Social link inference), a representation learning method customized for unsupervised learning scenarios. First, I regard trajectory data as a directed bipartite graph of users and POIs. Users and POIs are connected if there is a trajectory record between them. Features of an edge between users and POIs are defined by the aspect of each trajectory from additional information such as time slots, geographic information, and activity categories. I decompose the graph into aspect-wise bipartite subgraphs, each of which is learned separately by a context-aware embedding scheme. The overall structure of ANES is depicted in Figure 2.2.

One main point of the context-aware embedding scheme is that if a user visits a POI in a certain context, they should be close to each other in the embedding space by having a similarly transformed vector from their context. Different from previous approaches, which only focused on co-visitations of users, I learn the behavioral characteristics of users by considering contexts between users and POIs. First, user embedding is projected by each context to model the user's behavioral representation. After that, I define a transition-based relation between the projected user and the POI. In the context-aware embedding scheme, relations between users are modeled as a mixed form of projection and transition. Users' complex behavioral patterns in various contexts can be learned in this method. Context-aware embedding can lead to accurate representation learning useful for inferring hidden social relations between users without explicit label information in advance. After that, I measure the probability of social relations by comparing a pair of user embedding by concatenating the aspect-wise user embedding together.

ANES works well in the unsupervised condition by effectively defining loss function from the relation between users and POIs, to model complex behavioral patterns of users upon various aspects.

The major contributions of my work are as follows:

- I consider the influence of various aspects on the check-in data of users and propose ANES, an aspect-oriented social link inference framework. To the best of my knowledge, it is the first unsupervised solution that considers a projection-based embedding scheme on the social link inference problem.

• ANES decomposes check-in data into multiple aspect-wise bipartite graphs and learns representation of users according to their relations in each graph to fuse different aspects and contexts.

• I performed extensive experiments on four large-scale real-world LBSN trajectory datasets. The results show that ANES outperforms state-of-the-art baselines in predicting social links.

The organization of the paper as follows. In Section 2, I review previous research on network embedding methods. Section 3 defines the trajectory network and formally state the social link prediction problem. In Section 4, I present ANES with the general concept with algorithms as well as learning procedures from LBSN data. Section 5 describes the performance experiments. I use eight public LBSN dataset and compare ANES with state-of-the-art approaches. Ablation studies and the effect of parameters are also explained in Section 5. Lastly, conclusion and future work are included in Section 6.

2.2 Related Work

2.2.1 Location-based Social Networks

There have been a lot of researches aiming for social link prediction with the emerging interest of LBSN. Many previous studies used real dataset obtained from operational LBSNs such as Foursquare, Twitter, and Gowalla [39]. [12] proposed a social-historical model to consider the effect of social ties on check-in behavior. Spurred by the success of network embedding approaches [33, 41] in various domains, network embedding is intensively considered for handling trajectory data. One notable example is

GE [46]. GE applied PTE (Predictive Text Embedding) [40], originally proposed for embedding bipartite networks, for POI recommendation tasks. [19] used richer supplementary information in addition to data from LBSN. They collected WiFi access logs of students' smartphones and learned embedding by decomposing data to bipartite graphs and learning each graph iteratively.

Predicting implicit social networks such as friends from trajectory data also has gained much attention. [45, 26, 20, 13, 14] aimed to infer social relations between users from extracted trajectory data. Especially, [26] chose vehicle mobility to extract features of human behavior and utilized them for inferring social relations between drivers. [45] first applied Graph Convolutional Network (GCN) [25] to learn the embedding of trajectory and member of location-based social networks. My work is closely related to translation based knowledge graph embedding, which is known to learn the embedding of entities and relations effectively [4, 44, 27]. Those methods assumed that there is a translation vector for a relation when related entities are represented in an embedding vector space. In user trajectory data, the relations are visitation contexts (time, location) and entities are users and POIs. Representation learning schemes in recent studies are defined as embedding directed relations between users and POIs. One remarkable approach is STAE [35]. They utilized one of the translation-based knowledge graph embedding methods in POI recommendations. They also added a mechanism to effectively handle cold-start problem. However, few researchers have focused on using social network structure for the recommendation or for the link prediction using knowledge graph embedding methods. [54] proposed a multi view mixture model, which unifies location matching, time-series matching, and relation matching modules to unify those aforementioned points to improve the link inference. Recently, [56] proposed a hybrid personalized and neighbor attention model for link prediction, by considering local and global interest semantics. [23] considered privacy-preserving methods while considering geographical and social influence between users in the POI recommendation task. [24] proposed a holistic representation learning method to capture both social networks and check-in activities by splitting POIs.

2.2.2. Network Embedding

GCN (Graph Convolutional Network) [25], GCN and its improved schemes have been popularly adopted for network and node embedding. They aimed to learn graph structure more effectively or make it eligible for more complex graphs such as information networks and heterogeneous networks. Since Graph Attention Network (GAT) [42] has introduced the attention model in the GNN framework, some research adopted GAT onto various tasks in graph embedding tasks. EGNN [17] applies an improved version of GAT and GCN to exploit multi-dimensional edge features by defining an aggregation operation with edge features. MARINE [11] proposed a unified embedding framework to model both homogeneous and multi-relational heterogeneous graphs.

ASPEM [37] and PolyDeepWalk [28] proposed to define multiple facets for a single node to model heterogeneous relations between nodes. [32] first proposed a graph-attention-based embedding model for relation prediction in knowledge graphs (KG). They defined an attention operation for triplets consists of two entities and a relation and aggregated them to learn embedding of entities and relation. HetSANN [21] considered type-aware graph attention layers to encode the structural information of heterogeneous information networks. Heter-GCN [45] used R-GCN [36] to learn the visitation relation and co-occurrence relation of users and POIs. R-GCN proposes an advanced form of GCN aiming to handle graphs with multiple edge types by defining a different message-passing scheme for each relation. [22] considered multiplex networks which consist of several layers derived from different views and proposed three attention-based methods to fuse them for link prediction. [6]

regarded link prediction problems in graphs as a node classification problem in line graphs to minimize the information loss which occurs in graph pooling of traditional deep learning models.

2.3 Location-based Social Network Service Data

I use four large-scale LBSN datasets: Foursquare[1], Gowalla[2], Brightkite[3], and Yelp[4] public review dataset. For each dataset, I selected two cities to analyze the effect of geographical and social factors. All datasets contain user ID, POI ID, and geographic information of POI such as latitude and longitude. In addition, the common context information, each LBSN dataset

---

1 https://sites.google.com/site/yangdingqi/home/foursquare-dataset

2 http://snap.stanford.edu/data/loc-Gowalla.html

3 http://snap.stanford.edu/data/loc-Brightkite.html

4 https://www.yelp.com/dataset

provides additional distinctive information. For example, Foursquare data contains the check-in timestamp and the category of each POI while Gowalla and Brightkite data contain the timestamp, and Yelp data contain the category of each POI. I select users in the largest connected components in each dataset for a fair comparison with random walk-based models. Considering the different characteristics of Yelp and other datasets, I exclude the users with fewer than 10 reviews in Yelp data. Table 2.1 shows the statistics of the datasets.

Table 2.1. Statistics of dataset. Gowalla and Brightkite do not contain category data.

Dataset	Foursquare		Gowalla	
Region	NYC	Tokyo	Austin	Chicago
#Users	4,703	7,379	6,723	2,135
#POIs	30,340	59,802	15,866	6,867
#Check-ins	185,333	619,585	297,793	39,619
#AvgLinks	5.11	14.01	11.19	4.88
#Categories	410	404	-	-
Dataset	Brightkite		Yelp	
Region	Chicago	NYC	Portland	Vancouver
#Users	597	792	19,785	8,994
#POIs	3,561	5,872	17,432	12,832
#Check-ins	24,336	29,553	1,480,464	752,778
#AvgLinks	7.26	7.96	8.66	7.77
#Categories	-	-	2,290	2,054

In each dataset, I utilize check-in timestamps, geographic information of POIs, and categories of POIs as contexts if they are available. As timestamps of check-ins are not discrete data, I divide the 7 day 24 hour time interval into 168 time slots, which means that each time slot has a 1-hour length. For geographic information, I divide the region of each city into a 10x10 grid and designate each number of a grid that the POI belongs to. As shown in Table 2.1, my experiment is applicable to the inference of social relations in various types of trajectory data. Also, each dataset has a different number of users, check-ins, or average links between users. Experiments on those different datasets provide a good opportunity to examine the generality of ANES. The Portland-Yelp dataset has the largest user population and constructs the densest graph, while the Brightkite dataset are relatively sparse.

My work aims to infer social links using the trajectory data where each check-in is represented as a tuple of (user, contexts, POI). Contexts of each check-in are additional information which is contained in the data. From the check-in data, I extract the behavioral characteristics of users and find the most probable entities who are likely to have social links with them. In my environment, users with similar embedding have a higher probability to have a social link with each other. In LBSN data, I define the social link as shown in Definition 2.1.

Definition 2.1. A social link between a pair of users (u, v) is a reciprocal relation between them.

For the set of user ids U, the set of check-ins S, the number of aspects as n, and the set of POI ids as P, check-ins and aspects are defined as below:

Definition 2.2. A user's check-in $s = (u, p, c) \in S$ is defined when a user $u \in U$ visits a POI $p \in P$ with a set of contexts $c = \{c_1, c_2, \dots c_n | c_1 \in C_1, c_2 \in C_2, \dots c_n \in C_n\}$ From the check-in data, I construct n bipartite graphs $G_i = (V, E, C)$ such that

$$V = U \cup P\ , E = \{(u, p) | \exists c : (u, p, c) \in S\}, C = C_i.$$

In G_i, each check-in is represented as directed edge $\{e\} \in E$ with an edge feature $c \in C_i$. My objective is to learn the representation of each node in G to efficiently model the check-in behaviors between users, times, and POIs. In short, this paper provides answers for the research questions below:

RQ 1. Given trajectory check-in data as instances, how can I model relations between each element?

RQ 2. How can I learn the representation of each user from the heterogeneous network to utilize in the social relation inference task?

RQ 3. After learning representations of users, how can I measure the probability of social links between users?

2.4 Aspect-wise Graph Decomposition

To model the behavioral characteristics of users in LBSN, I propose an aspect-wise decomposition scheme to consider various aspects of check-ins. As one check-in record contains multimodal information such as geographic, temporal, and spatial features, I build subgraphs based on each of those features. In Figure 2.2, I show an example of two aspects in an LBSN graph. Two subgraphs have the same nodes and edges, but their embedding and edge features are different. All subgraphs are passed separately to a context-aware embedding scheme.

Note that the number of aspects varies on each dataset. For example, user behavior from review data may not contain the exact visitation date, which is crucial for detecting co-visitations. Therefore, it is inadequate to practice co-visitation based methods to such data. However, ANES can learn the embedding of users and POIs utilizing any available aspects such as categorical and geographical data. This point makes ANES more flexible and generalize better to any data that contain a plethora of visitation and behavioral contexts.

Algorithm 1: Learning ANES in a single aspect

Input : *Trajectory instances T{(u,p,c)}, mini-batch size k*
Output: Embedding for users E_u, contexts, E_c, M_c and POIs E_p
Ini$tialize$ E_u, E_c, M_c, and E_p (Normal Xavier Initialization)
Sample a mini-batch *s* of size *k* triples from *T*
for each epoch do
    for each minibatch *s* do
        for *(u,p,c)* in *s* do
            (u',p',c')← *corrupt(u,p,c)* // corrupt each triple
            Calculte *L(u,p,c)* according to Equation (2)
        end for
        Sum up *L(u,p,c)* and update each embedding w.r.t Equation (3)
    end for
end for

2.5 Aspect-wise Graph Learning

The key idea of ANES is to uncover users' behaviors which change depending on contexts. In the *i*-th aspect, I define the transition-based relation between POIs and Users of certain context ci as follows.

$$M_{c_i} e_u^{(i)} + e_{c_i}^{(i)} \cong e_p^{(i)}$$

where $e_u^{(i)}$ and $e_p^{(i)}$ are embedding of a user u and a POI p in i-th aspect, respectively. M_{c_i} is a translation matrix that maps the user space to the POI space under the context, c. If u visits p in context c_i, embedding of u on context c_i is denoted as $M_c e_u^i$. Also, there exists transition embedding vector $e_{c_i}^{(i)}$ between $e_p^{(i)}$ and $M_{c_i} e_u^{(i)}$. After that, I define an aspect-wise score function of a triple (u, p, c_i) as dot product of transformed user embedding and POI embedding as follows.

$$score(u, p, c_i) = (M_{c_i} e_u + e_{c_i}) \bullet e_p$$

Semantically, this score function depicts how a user's embedding in a certain context has a transitive relation with her visited POI. This approach can provide sufficient representation space to learn the information of trajectory data on LBSN.

To train ANES, I aim to maximize the difference between the score of real triples and the scores of arbitrary fabricated samples. For a real triple (u, p, c_i), I propose the following contrastive loss function.

$$L(u, p, c_i) = -log(\sigma(score(u, p, c_i))) - \sum_{(u', p', c_i)} log(\sigma(-score(u', p', c_i)))$$

where (u', p', c_i) is an arbitrarily fabricated check-in instance. In equation 3, I adopt a negative sampling method inspired from [9]. ANES generates negative samples from existing triples by 'corrupting' a part of triples. When generating negative samples, I randomly pick users or POIs according to the distribution $P(u)$ and $P(p)$, which are proportional to the 3/4 squares of count in data. The loss function increases the score of positive triples while lowering the scores of negative triples samples. A sigmoid function for σ prohibit steep increases and decreases of positive triples and negative triples, respectively. For multiple aspects, the sum of aspect-wise loss functions should be minimized. I define the objective function of ANES as follows:

$$O = \sum_{(u,p,c)\in S} L(u,p,c_i)$$

As I use mini-batch as the training procedure, I train ANES from the sum of all losses for given triples of each mini-batch. The procedure for training ANES is described in Algorithm 1.

2.6 Inferring Social Relation from User Representation

After embedding are learned for each user, time, and POI, I concatenate users' embedding on all aspects as representations for their behavioral patterns.

$$e_{u_i} = Concatenate(e_{u_i}^{c_1}, e_{u_i}^{c_2}, \dots e_{u_i}^{c_n})$$

From learned representation of users from each aspect, I compute the pairwise similarity of user embedding as a probability that a social link exists between the user pair. In my experiment, cosine similarity is used.

$$P\,(u,v) = similarity(e_u, e_v)$$

I compare ANES with seven state-of-the-art methods which can be performed in an unsupervised setting. The selected baselines are the best performing methods representing different approaches; graph embedding, knowledge graphs, co-visitation, and multi-context embedding. I use the same hyperparameter values indicated in baseline papers. The baselines are as follows.

• Deepwalk [33]: Deepwalk performs random walks and skip-gram learning to gain the embedding of each node. I apply Deepwalk on the User-POI graph ignoring edge features since Deepwalk does not support edge features.

• Walk2Friends [1]: Walk2Friends is a skip-gram-based model which is proposed to be used on a user-location bipartite graph for social link prediction.

• Metapath2vec++ [9]: Metapath2vec++ is a meta-path-based representation learning model specialized for heterogeneous graphs. To distinguish between Metapath2vec++ and deepwalk, I construct a tripartite graph of users, time slots, and POIs. Also, I build meta-paths from the tripartite graph. In that case, all contexts in each aspect are also considered as nodes.In Yelp data, I construct a graph using categories, instead of time slots.

• STA-E [35]: STA-E is a KG-based embedding method that considers spatio-temporal contexts of each user using TransR [27]. I choose it to examine the possibility of adopting KG embedding methods for social link inference. Note that STA-E is devised for POI prediction. However, I include STA-E in the baselines to investigate the impact of knowledge graph embedding.

• Heter-GCN [45]: Heter-GCN is a seminal representation learning model that first applies GCN for social link prediction. Heter-GCN uses R-GCN [36] to extract embedding from the user-location graph. Heter-GCN utilizes Skip-gram and Gated Recurrent Unit [7] also to pre-train embedding [57] before passing it into GNN. For pre-training, I use the same parameter values suggested in [57, 45]. Because Heter-GCN utilizes visitation time slots solely, I cannot apply Heter-GCN to the Yelp dataset and its performance result is not included.

• PolyDeepwalk [28]: Similar to ANES, PolyDeepwalk considers multiple facets of nodes to represent the rich semantics of networks. I concatenated the overall representation of all facets to evaluate the performance.

• ConvE [8]: ConvE is a convolution-based knowledge graph learning model. It defines the scoring function as a combination of convolutional and fully-connected layers. To make a fair comparison, I considered all scenarios which include check-ins similar to Table 2.2 and stated the result with the best combination of features.

2.7 Performance Analysis

In each dataset, I utilize check-in timestamps, geographic information of POIs, and categories of POIs as contexts if they are available. As timestamps of check-ins are not discrete data, I divide the 7 day 24 hour time interval into 168 time slots, which means that each time slot has a 1-hour length. For geographic information, I divide the region of each city into a 10x10 grid and designate each number of a grid that the POI belongs to.

As shown in Table 2.1, my experiment is applicable to the inference of social relations in various types of trajectory data. Also, each dataset has a different number of users, check-ins, or average links between users. Experiments on those different datasets provide a good opportunity to examine the generality of ANES. The Portland-Yelp dataset has the largest user population and constructs the densest graph, while the Brightkite dataset are relatively sparse. I use 128 or 256-dimensional vectors for embedding of users, POIs, and transition matrix of each aspect. All embedding vectors are initialized using Xavier initialization [16]. Also, I use 128x128 or 256x256 matrix for the projection matrix. I adopt the Adam optimizer with a learning rate of 0.0001. The number of negative samples per positive sample is fixed to be five. To evaluate the result of embedding from ANES and other baselines, I split the social relation data into 20/80 for the validation set and test set, respectively. The batch size for learning ANES is set to be 128 for all datasets. The maximum epoch is 5,000. L2 regularization for all parameters is applied with a weight decay coefficient of $5 * 10{-}4$. ANES and other baselines are learned and compared on the server with a single TITAN X GPU.

I learn a low-dimension representation of users and calculate the similarity of each embedding to generate a score that indicates the probability of a link between two users. Because each baseline uses its own similarity measure methods, I use three metrics - dot product, cosine similarity, and Euclidean distance - and select the best result among the three. As my approach is evaluated in an unsupervised environment, I use Area Under the ROC Curve (ROCAUC) as a performance metric.

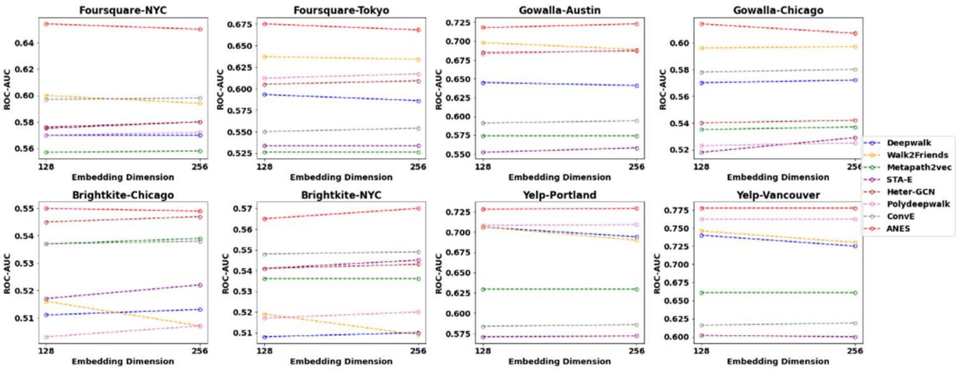


Figure 2.3. Performance comparison of ANES in 8 LBSN dataset.

Figure 2.3 compares the performance of ANES and those of the seven baselines. The results show that ANES performs better than other baseline approaches in all eight datasets in both embedding dimensions. As ANES considers the overall behavioral patterns of users in different contexts, it shows better inference than other graph based, or co-visitation based GCN methods. Also, ANES shows superior performance than multi-context and knowledge graph-based methods.

One notable observation is that Walk2Friends is ranked second in case of Foursquare and Gowalla datasets. I guess that in large graphs such as Foursquare and Gowalla, the mechanism of averaging the representation of neighborhood nodes may cause the loss of information. However, on a small Brightkite dataset, Heter-GCN performs better than other random walk-based methods. Also, compared to Heter-GCN, as the number of check-ins increases, there is a higher chance of opportunistic co-visitations among users without social relations. This may deteriorate the performance of Heter-GCN by building many irrelevant edges in co-visitation graphs. To investigate this further, I discuss the relationship between data sparsity and the performance of social relation inference in Section 2.8. On the Yelp datasets that do not have timestamp information, Polydeepwalk shows the second-best performance. It implies that multi-context learning is effective on categorical check-in data. In contrast, knowledge graph-based methods do not perform well, suggesting that they fail to efficiently extract categorical information. ANES considers multiple contexts in each aspect and shows its effectiveness and superiority in social link inference.

2.8 Discussion and Implications

In this section, I analyze the findings from my results in a greater detail to shed light on the fundamentals of ANES.

2.8.1 Distribution of Social Links

As mentioned in previous sections, traditional approaches have focused on the co-visitation of users. However, they ignore propagated relationships which play an important role [52] in predicting social relations. I classify each social link by the closest interval of co-visitation and check how ANES effectively predicts them. In Figure 2.4, I divide social links by their minimal co-visitation interval. ANES, designed for inferring relationships beyond co-visitation, shows a notable increase in performance when predicting user pairs that have no co-visitation. It reflects that ANES learns the propagated relationship more effectively than other graph-based approaches. Also, ANES showed the best performance when co-visitation interval is less than 1 hour. It implies that my definition of time slots and my approach to learn visitations by time slots is effective.

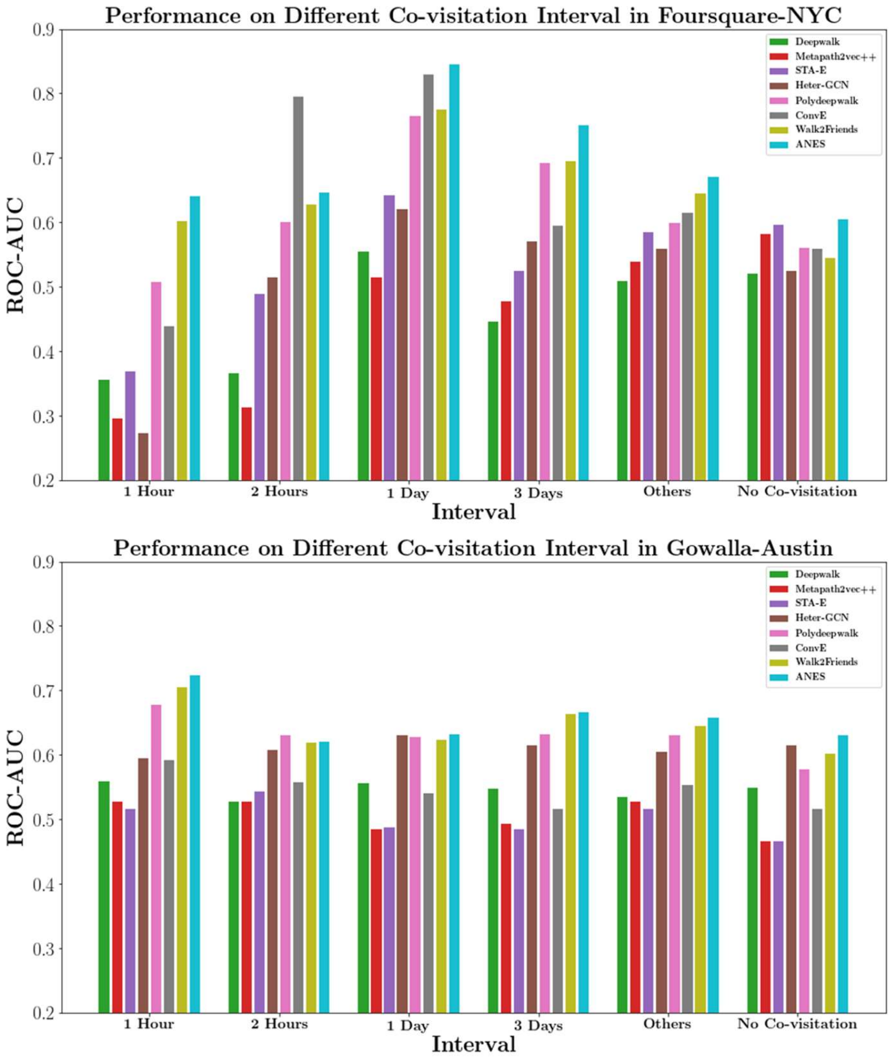


Figure 2.4. Performance of social relation inference at different co-visitation intervals. ANES effectively infers social relation in both situations when there is no co-visitation between user pairs and when the co-visitation interval is within 2-hours.

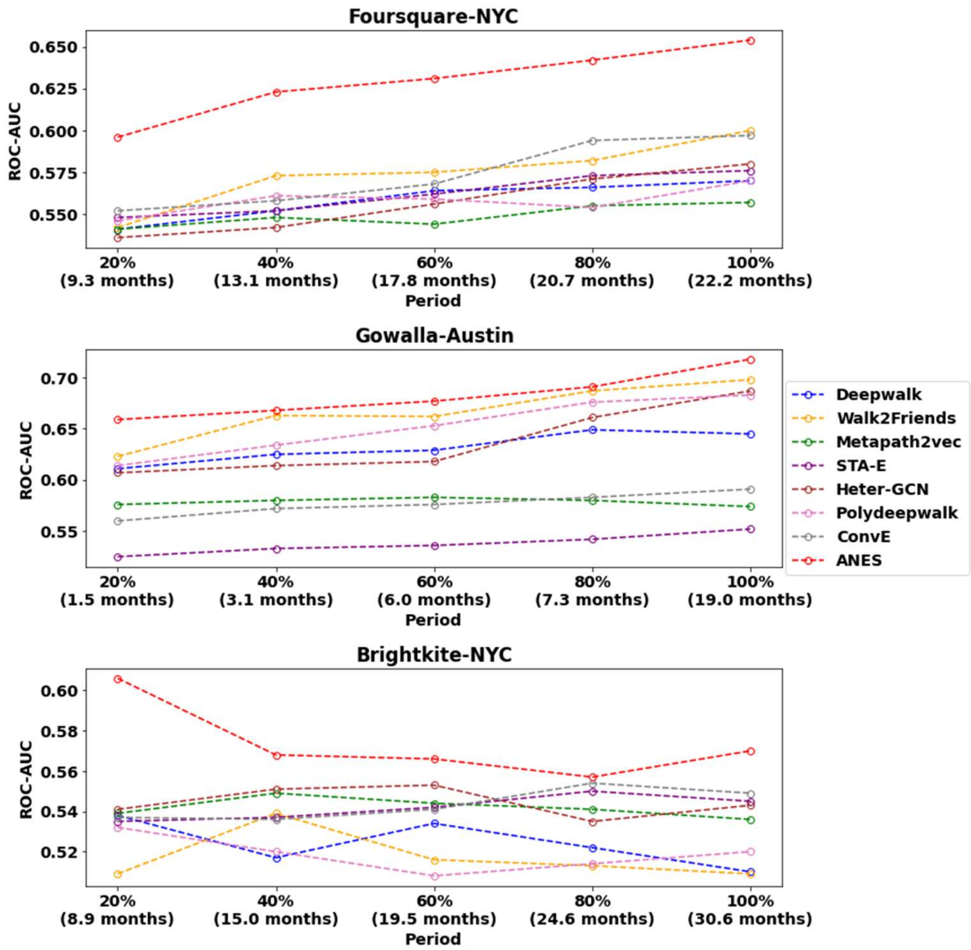


Figure 2.5. Performance on different proportion of data on three datasets

2.8.2 Effect of Data Scarcity

I compare the performance of social link inference by varying the scarcity of check-in data. Many existing random walk or graph neural network based approaches utilize graph structure, and their performance can be affected by the degrees or sparsity of graphs. It is an important issue in the social link inference problem because predictions at initial stages while data is insufficient is important. To investigate the issue, I do the same task while changing the proportion of check-in data from 20% to 100% and apply ANES in the reverse chronological order from the last day of the check-in data. Note that I only evaluate the data for users who have at least one check-in during the training period. Figure 2.5 shows the range and the result on five different proportions of LBSN data.

Compared to the baselines, ANES shows the best performance in all experimental settings. Especially, ANES infers the social relationship of users better when the amount of provided data is small. When I use 20% of the dataset, ANES shows an average of 8.59% improvement than the second-best baselines.

A notable observation is that the relative ranks of baselines are different on each dataset. Heter-GCN performs better only in Brightkite-NYC data where the number of users is smaller than other datasets. There are noticeable differences in the characteristic of the Gowalla-Austin dataset and that of the Brightkite-NYC dataset. Because the Gowalla-Austin dataset contains a richer record of visitations accumulated in a short period, the temporal difference between check-ins can be ignored. This characteristic leads to good performances for time oblivious schemes such as random walk. Also, ANES performs well because it considers multiple aspects in addition to temporal information. On the contrary, on the Brightkite-NYC data where check-ins are sparse, Heter-GCN's pre-training scheme discriminates 'old' check-ins from 'new' check-ins and efficiently extracts information that may change over time.

Another interesting point is that Deepwalk and Metapath2vec perform better on the 80% data of the Gowalla-Austin dataset than on the 100% of the data. Because the time interval increases from 7.3 months to 19.0 months while the size of dataset increases only 20%, I can contemplate that additional check-in data are quite sparse and Deepwalk and Metapath2vec may not handle the sparse Gowalla-Austin dataset adequately. As the two random walk based models could not model each visitation differently, they seem to be vulnerable to noisy data from the early period of overall data. On the contrary, because ANES does not use random walk or GNN, they show robust performance regardless of whether the data or graph are dense or sparse. Those observations imply that ANES can be applied faster and more quickly than other baselines, with a small amount of data.

2.8.3 Effect of Aspects

In this experiment, I analyze the performance of ANES applying only subsets of aspects to investigate the impact of each aspect. As I consider time slot, categorical, and geographical data as aspects, I denote them as T, C, and G. Table 2.2 shows how aspects affect the performance of ANES. The result indicates that the geographical aspect is the most important in gaining the performance. The temporal and categorical data also boost the performance when they are combined with the geographical aspect. It implies that the graph decomposing scheme of ANES is effective to put all features together.

Table 2.2. Performance of ANES with different subsets aspects. T, C, and G denote time slot, categorical data, and geographical data, respectively.

Dataset	Foursquare	Gowalla	Brightkite	Yelp
Region	NYC	Austin	Chicago	Portland
ANES (TCG)	0.654	-	-	-
ANES (TC)	0.633	-	-	-
ANES (TG)	0.645	0.719	0.550	-
ANES (CG)	0.643	-	-	0.729
ANES (T)	0.615	0.683	0.539	-
ANES (C)	0.598	-	-	0.678
ANES (G)	0.636	0.705	0.547	0.716

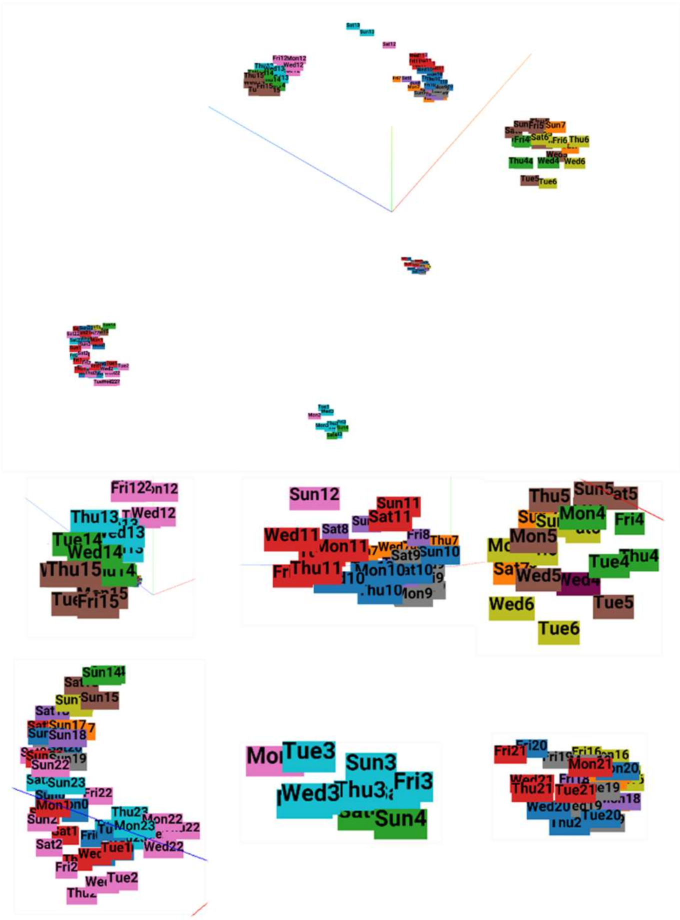


Figure 2.6. Visualization of time slot embedding using t-SNE. The upper figure is the projected position of each relation embedding in 3-D space, and the lower figure is an enlarged version of each embedding to show which relations are close to each other.

2.8.4. Visualization of Time slots

Here, I visualize the learned representations of time slots using t-SNE [30]. I parameterized each time slot by M_t and E_t. Figure 2.6 maps M_t onto three dimension spaces. I omit E_t because its mapping is similar to Mt. In

Figure 2.6, I can observe that 168 time slots are divided into six clusters. Each cluster represents a set of time slots that have similar embedding. The Figure illustrates that cluster boundaries are roughly related to weekday/weekend and time-frame in 24 hours. As I mentioned in previous sections, I modeled different projections for time slot because the same behaviors during different context can defer. Also, users' behaviors are periodic with a 24 hour cycle. The first cluster at the upper leftmost corner represents activities around lunch on weekdays. During that time slots, users go out for lunch with friends or colleagues. The second cluster at the center upper is similar to the first cluster, but it is related to the morning. Likewise, other clusters can be regarded as time slots in dawn or evening. The third and fifth clusters represent late day or early evening, and check-ins are not as frequent as other time slots. Also, weekday and weekend time slots are mixed together at the third and fifth clusters. The fourth and the sixth cluster consists of weekday night slots and weekend daytime slot, indicating that people tend to stay at home during those time slots. In short, the Figure 2.6 implies that my time slot based ANES model properly models each time slot without any pre-trained embedding.

Furthermore, I show top 10 similar/dissimilar pairs of timeslots by calculating cosine similarity of M_t in Table 2.3. In Table 2.3, timeslot with 10 o'clock in a day have highest similarities among all pairs. It implies that check-in behaviors of users are almost equivalent in each day. In dissimilar pairs of timeslots, many are calculated between weekday and weekend, and morning and afternoon or dawn.

Table 2.3. Top similar/dissimilar pairs of timeslots

Timeslot Pair	Similarity	Timeslot Pair	Similarity
Mon 10 – Tue 10	0.8316	Tue 14 – Sat 2	0.1667
Mon 10 – Wed 10	0.8177	Sun 3 – Mon 16	0.1700
Tue 10 – Fri 10	0.8165	Sun 1 – Tue 14	0.1710
Mon 10 – Fri 10	0.8154	Mon 0 – Mon 14	0.1724
Tue 10 – Thu 10	0.8065	Mon 14 – Sat 2	0.1764
Mon 10 – Fri 10	0.8050	Sun 1 – Fri 14	0.1770
Tue 11 – Thu 11	0.8037	Sun 1 – Wed 15	0.1798
Tue 9 – Fri 9	0.8008	Sun 3 – Tue 14	0.1839
Tue 10 – Wed 10	0.7963	Sun 0 – Fri 14	0.1844
Tue 11 – Fri 11	0.7927	Tue 14 – Sat 23	0.1848

2.9 Summary

In this paper, I propose ANES to perform representation learning on User-POI check-in data for unsupervised social link inference. I design the aspect-oriented relationship between users and POIs to learn both the pairwise relation and network structures. I show my approach is superior to other state-of-the-art methods by showing the result of the unsupervised link prediction task. I also analyze the effect of each aspect in various conditions on the real datasets. ANES can be applied to any location-based domain because it does not need any pre-training and additional information on existing social relations. Also, contrary to the co-visitation based methods which requires check-in timestamps, ANES provides adequate performance using other available aspects such as geographical and categorical information.

While I define an effective way to decompose check-in data into multiple subgraphs, there is still space to improve in ways of learning those subgraphs. I would expect improvement as I can extract more information from check-in data and learn the graph effectively by considering the following points.

- As I prove that ANES is powerful to infer social relationships between users in a relatively short amount of data collection, I will explore the dynamics of social link formation such as predicting relations that are likely to be disconnected in the future.
- As multi-aspect embedding of ANES has proven to be effective, I will seek a way to expand my work to other problems beyond friend inference.
- Finally, I will improve ANES to consider the chronological relation between check-ins in the future. This improvement may include dividing graphs by timestamps or considering time-related relations between check-ins.

Chapter 3 Detecting collusiveness from reviews in Online platforms and its application

3.1 Background

While I want to think that purchasing products in online-markets is mostly based on my own decision, by many routes my preference and desire for everything can be easily influenced by others' opinions. From the reviews of new hotels or restaurants to the Tomatometer of Rottentomattoes, my decision-making process are closely related to the experience of previous customers. In 2019, 91 percent of U.S adults responded that they read online reviews for new purchases, and more than 80 percent of review readers trust online reviews as much as a personal recommendation, according to the BrightLocal Consumer Review Survey[5]. Furthermore, about 41% of responders think that reading online reviews is helpful, which is more than 33% of responders who chose government oversight, according to Pew Research Center's 2016 survey[6] . That's why many online shopping platforms have exhibited their customers' reviews at the front of their product page [70] . This belief of customers starts from the assumption that all reviews they see are genuine and sincerely-written without frauds.

Unfortunately, there are almost no such naïve scenarios that only genuine customers write product reviews. Due to the importance of reviews on the purchasing process, many merchandisers or service-owners want to store positive reviews on their products' pages. To fulfill their desire quickly and easily, they tend to manipulate fake reviews, such as paid reviews which are generated to boost certain products, to mislead potential customers by hiding genuine consumers' opinions. There are lots of online services that can be easily found to aid 'marketing' of certain products, by providing 'genuine-alike' fake reviews on various platforms [90]. Amazon Mechanical

5 https://www.brightlocal.com/research/local-consumer-review-survey/

6 http://www.pewinternet.org/2016/12/19/online-reviews/

Turk is one of the popular websites in which purchasers can hire crowd workers to promote their products or demote competitive products.

Many approaches have aimed to detect certain suspiciousness from review data. Some research has focused on the contents of reviews, by utilizing personal insights [73,94,95]. Excessive use of unigrams or ALL-capitals words were examples of text-oriented features. Also, techniques to find reviews that are significantly different from other users have been developed. According to previous researches, opinion spam reviewers tend to show bimodal distribution on their posting rates [76] or tend to be intensively active in a short period [78].

However, malicious reviewers became harder to find, by disguising their writing style and hiding their abnormal behaviors by using compromised accounts [69]. Therefore, a central point of detecting so-called ‘Sybil’ spammers are changing to consider the objective of opinion spamming itself. Even though the spamming account protects themselves, they should come out of the closet for their purpose to manipulate some products’ status. For that reason, their behavior should be collusive. My research has mainly focused on this characteristic of opinion spammers.

In this paper, I propose SC-Com, a robust spam detection method which can be used in many domains by only considering primary data such as time data and ratings. SC-Com has caught opinion spam reviewers effectively and precisely in large-scale real-world data. It aims to extract communities of reviewers with their behavioral collusiveness and utilizes features from each community to supervised learning for detecting opinion spammers. First, I observe the importance of finding collusiveness in opinion spam, by constructing a projection graph of reviewers according to their behavioral similarities. Second, I propose the framework to divide whole users into closely related communities and discriminate each community by extracting features related to communities' suspiciousness. Lastly, I prove that my approach can provide a reliable and robust solution with a high F1 score of 0.93 in a real-world dataset. Figure 3.1 shows an overview of my proposed framework.

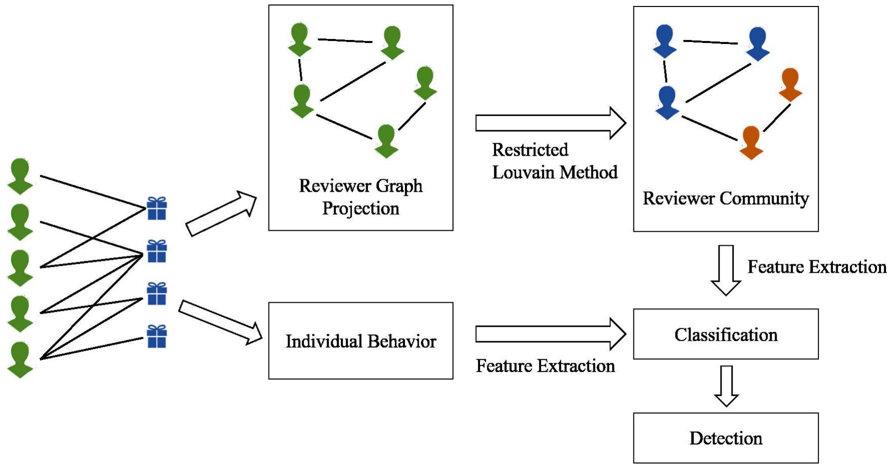


Figure 3.1. SC-Com utilizes community-based data from user similarity projection graph and review data to classify reviews as spams and benign reviews.

The main contributions of this paper are summarized as follows:

• I propose SC-Com, which captures collusive spammer groups from the rating and time data, which are immune to the camouflaging of writing skills, and I provide a procedure to optimize my method for a real-world dataset.

• SC-Com is strong against camouflaging of spammers such as adding irrelevant reviews by restricting the weight of edges while detecting communities.

• From the experiment using the real-world opinion spam dataset, my method outperformed previous detection methods especially by showing the improvement in recall.

3.2 Related Work

Research in identifying spam reviews aims to find suspicious reviews which are likely to have different characteristics from regular reviews ('Benign'). Lots of features have been extracted and used for review spam detection. While many researchers have utilized users' behavior to discriminate spams from benign reviews, some are interested in finding groups of suspicious users, by finding collective behavior of malicious activities. I briefly introduce many approaches to detect spam reviewers in history.

3.2.1. Individual Spam Detection

One of the well-known insights is that spam review has malicious contents which are commonly regarded as 'paid reviews'. That was the reason why the dataset in traditional researches is revised with human investigation. For example, [94, 95] utilized their personal knowledge to label the review data, with the information of deleted review labels. From the computational approaches such as [73], some personal knowledge has been extracted as key features for opinion spam detection. For example, the ratio of capital words, the ratio of brand names, and the length of the review have been utilized for detections in [73,81,95]. Furthermore, finding differences in the review text of users using natural language processing has been proposed, and showed decent result [62, 64, 82, 92. However, malicious reviewers became harder to find, by disguising their style of writing. That is the reason why behavioral features have been considered in addition to linguistic features in opinion spam detection [81,95]. From the insight that spam accounts which are created for malicious behavior are active in an intensively short period, the maximum number of reviews in a day [78] or some metrics to describe the 'burstiness' of users have been proposed in [66]. It is known that many malicious behaviors in the social network are showing a suspicious pattern in the viewpoint of time distribution, and the temporal dynamics of spam users are different [72]. [68] applied a Bayesian-oriented approach to find opinion spam reviews using time series of user activities.

[81] proposed a notable example which considered text, behavioral, and network data altogether. They assumed that user, review, and product could be divided into two classes, which are 'spammer/benign', 'target/non-target', and 'fake/genuine'. From the initially calculated prior probabilities from data, they performed loopy belief propagation to classify each network object collectively. Similar assumptions are proposed in [75, 91]. [75] have proposed three different score metrics for each rating, user, and product as reliability, fairness, and goodness. Their assumptions were that there would be a proper score which should be given to a product if all users are genuine, and fraudsters tend to give low ratings to good products and give

high ratings to bad products. [77] considered a multimodal network embedding method using a probabilistic review graph to calculate the spamcity of review, reviewer, and products. [65] considered existing metrics for spam detection as priors and performs loopy belief propagation to ensemble those metrics. [99] considered recurrent convolutional neural network to discriminate deceptive review from other truthful reviews.

[87, 89] considered a way to solve imbalanced class issues, because the number of spam reviewers is quite smaller than benign users. They proposed a variation of support vector machine (SVM) or generative adversarial network (GAN) to overcome the scarcity of dataset.

3.2.2 Collusive Spam Detection

Many researches tried to find the collective behavior of malicious users because review spams are intended to hide honest reviews and dump positive/negative reviews to falsify the rating of products [71,72]. Their behavioral patterns should be collective and synchronized to dump many contents effectively. This approach has been widely used in social media such as finding socially similar accounts [61]. One example is bimodality and co-bursting, which is crucial in spam reviews. [76] focused on those two phenomena and proposed the Hidden Markov Model-based method. [69, 85] focused on finding groups of suspicious users according to their behavioral patterns and extracted those 'blocks' in the network. Those kinds of approaches were useful for many domains because only simple features as time sequences are needed to detect blocks from whole networks. [69] proposed a suspiciousness score for each node and edge and calculated the density for subgraphs by dividing the total score by the size of each subgraph. By removing less suspicious edges and nodes from the whole graph until the density is maximized, they could detect dense blocks iteratively. They provided improved detection results from the similar previous approaches of [101] and also provided theoretical insights to justify their greedy-based algorithm is powerful. [83] focused on identifying spammer ratings based on differences between spammer ratings and the majority ratings of honest reviews. [97] focused on a single particular product and proposed indicative signals for the products' time series of reviews. They checked the 8 indicative signals such as average rating and a ratio of singletons, and they detected the possibility of anomalies such as spamming activities from the sudden change of those signals. [74] inferred the hidden collusiveness between singleton spammers to detect group spam attacks.

From many researches, activities of users are often represented as graphs to detect collusive spams. [59, 93, 94, 99] proposed user-user projection graphs which are converted from bipartite review graphs. Some well-known community detection method has been used to catch malicious communities among the whole user network [80]. [99] have constructed suspicious social links from review data and employed the Louvain Method [60] to extract communities. After that, they tried to boost their detection techniques by adding the method to discriminate against Sybil communities. [79] utilize tripartite graphs, which consider reviewer, rating, and product as heterogeneous nodes. [96] focused on finding nested spammer groups and targeted products by ranking users and products by metrics such as abnormality and diversity, and they proposed Network Footprint Score (NFS) to guess suspiciousness scores of each product.

[63] proposed a notable approach which utilized graph embedding such as Node2Vec [67] to rank spammer groups based on their collusiveness. [84] utilized review spam features as heterogeneous information networks for a network classification problem.

3.3 Online Review Data

I use the dataset of [95], which is collected from Amazon.cn. It contains a snapshot of 1,205,125 reviews written by 645,072 reviewers on 136,785 products. I choose this dataset for validating my method because it has about 8-years long-term data and it is good to show general applicability by using one of the most popular e-commerce website data. Furthermore, due to the system that reviewers cannot delete their own review by themselves, labeled data is more reliable compared to datasets from other domains. Each review data has two-class labels, non-spammer, and spammer. 1,937 reviewers are manually annotated as spammers in total. Though original data has six attributes (Reviewer ID, Product ID, Rating, Date, Product Brand, and Review Text), I decided to use only four attributes to show the effectiveness of my proposed method without considering linguistic features.

To start utilizing online review data, I assume a set of a user $U = \{u_1, u_2, \dots u_n\}$ and a set of products $P = \{p_1, p_2, p_3, \dots p_m\}$. Each review has four features as {user, product, time, rating}, and each element means the ID of a user, ID of a product, posted time of a review, and rating on a review. A user u_i and a product p_j are connected if there is a review on p_j by u_i. Intuitively, they form a bipartite graph.

From the bipartite graph of reviewers and products, I now propose SC-Com. SC-Com starts from converting the review graph into a projection graph between users, and it performs a community detection procedure to group users by their similarity. The last part of this section is an explanation of the feature selection and classification method for supervised learning.

3.4 Collusive Graph Projection

Before grouping reviewers as communities, I project the bipartite graph G into the weighted undirected graph of reviewers based on their pairwise collusiveness. Pairwise collusiveness of two reviewers is defined as the sum of elementwise collusive scores which is calculated within one product. I choose each elementwise collusive score as an exponential form of feature difference, as the possibility of colluding should be maximized when those features are the same, and it should decrease fast as the difference increases. Four basic features are used to define elementwise collusive score in a single product, including:

Time Difference: The difference of time between two reviews in a product. Normally, the participants of spam campaigns tend to write reviews in a fixed period, so their collusiveness is bigger when they write reviews in a close time.

Rating Difference: The difference of review rating between two reviews in a product. Writing reviews with the same polarity is essential in opinion spam behavior.

Time Freshness: The time gap between their reviews and the first review of the product. I choose the later review between two reviews to calculate the time gap.

Rating Abnormality: The difference of review rating between the average rating of the product and two reviews. I get an average rating difference of average rating and two reviews. Using those four features, the pairwise collusiveness score function is defined as:

Definition 3.1. For a user pair (u, v), let p denote a product which is reviewed by both users. The pairwise collusiveness score $c(u, v, p)$ of u, v on p are defined as follows:

$$c(u,v,p) = e^{-\left(\alpha|t_{up} - t_{vp}| + \beta|R_{up} - R_{vp}| + \gamma[max(t_{up}, t_{vp})] + \delta\left[R_p - \frac{R_{up} + R_{vp}}{2}\right]\right)}$$

The four weight parameters α, β, γ, δ are indicative of the significance of each feature in relation to one another, with higher values indicating greater importance.

To optimize four parameters, I aim to define new term which is calculated by ratio of collusiveness scores of collusive spammer pairs among all pairs of reviewers in a product. I call the term as Well-formness as follows:

Definition 3.2. In each product p, the Well-formness $W(p)$ is the ratio of the sum of collusiveness scores of positive colluding pairs (u', v') to the sum of collusiveness scores of all pairs (u, v).

$$W(p) = \begin{cases} \dfrac{\sum_{(u',v')} c(u', v', p)}{\sum c(u, v, p)} & if\ there\ exists\ a\ colluding\ pair\ (u', v') \\ 1 & otherwise, \end{cases}$$

To jointly maximize the Well-formness in all products, I propose the target function O as follows:

$$O = \prod_{p} W(p)$$

To maximize the target function, I adopt the Stochastic Gradient Method (SGD) to optimize all 4 parameters. Optimized parameters and their implications are stated in Table 3.1. Table 3.1 provides an insight to understand which parameter is more important compared to others. As α is bigger than γ, the impact of time differences drops faster than the impact of time freshness as it goes far from zero.

Table 3.1. Values of optimized parameters in SC-Com.

Parameter	Value
α (Impact of time difference)	0.001484
β (Impact of rating difference)	0.717043
γ (Impact of time freshness)	0.000266
δ (Impact of rating abnormality)	0.516693

From the pairwise collusiveness score of product p for two users, I add them cumulatively for all products to derive the overall collusiveness of two users.

Definition 3.3. The collusiveness score $C(u, v)$ of two users (u, v) is the sum of pairwise collusiveness scores.

$$C(u,v) = \sum_{p} c(u,v,p)$$

Therefore, if two users have written collusive reviews in several products together, the weight of the edge between them would probably be higher than others. One of the strong points of my definition of collusiveness score is that the similarity is not decreasing by camouflaging. Even if a spammer writes lots of genuine-alike reviews to pretend to be a normal user, the collusive score between him and his colluders are not affected.

3.5 Reviewer Community Detection

From the constructed projection graph of 3.1, I divide the whole projection graph as subgraphs, by performing Restricted Louvain Method, which is a modified version of one of the most widely used community detection algorithms. While Louvain Method has been outperformed empirical community detection methods in modularity, it is not efficient to use this algorithm directly in my constructed similarity graph. That is because the weight of edges is not standardized, so the classic Louvain Method would construct large communities which consisted of some non-spammers and spammers mixed. To get a result of more partitioned communities, I adopt a threshold term λ to discriminate between heavily collusive users and others. My Restricted Louvain method based community detection algorithm is shown in [Algorithm 2].

Algorithm 2: Restricted Louvain Method

```
1 Input : G =(V,E),
2 Output: Community designated for each node I
3 Initialize q = −∞
4 while q < new modularity do
5    Put each node in G in its sole community
6    while stabilized do
7       for all node n of G do
8          for all neighbor i of n do
9             if weight(i,n) < λ : continue
10            put n into its neighbor's community which maximizes the modularity gain
11         end for
12      end for
13   end while
14   Update q as new modularity
15   G = the network between communities of G
16 end while
```

In line 9, I compare the collusiveness of nodes with their neighbors and prohibit the case of merging communities with small collusiveness. I could make more strongly connected communities whose node has at least one edge with a collusive score which is higher than λ with its neighbors. The selection of parameter λ in [Algorithm 2] is related to the direction of forming community structure. If λ has a high value, I restrict the minimum weight of edge more to construct many small communities. If λ has a small value, big communities would likely be formed. I investigated the impact of choosing appropriate λ by varying it in Section 3.8.5. After this section, all experimental results are gained when λ=1.7, which are shown as best in my experiment.

From each community, their network structure is a useful feature revealing that they are participants in a campaign. Table 3.2 shows the selected list of community-related features I used in the classification. Especially, I used entropy-related features which are inspired by [81], such as CF8-CF10. The full list of features I used is in [Appendix 1]. To measure how a community's members' collusiveness, I consider reviews written by them together to check the collectivity of rating and review time interval.

Table 3.2. Features extracted from detected communities. Shared features mean all members in a community have the same value, while Individual features are different individually. My original features are noted as *.

Number	Feature	Category
CF1	Number of nodes in a community graph	Shared
CF2	Number of edges in a community graph	Shared
CF3	Density of a community graph	Shared
CF4	Average degree of a community graph	Shared
CF5	Number of products reviewed by members of a community	Shared
CF6	Number of reviews written by members of a community	Shared
CF7*	Entropy of number of reviews by products	Shared
CF8*	Entropy of intervals of writing time of a community	Shared
CF9*	User Entropy of Review Count	Shared
CF10*	Entropy of Rating Distribution	Shared
CF11	Average Time Difference	Shared
CF12	Standard Deviation of Time Difference	Shared
CF13	Average Rating	Shared
CF14	Standard Deviation of Rating	Shared
CF15	Lifetime of a community	Shared
CF16	Average weight in a community graph	Individual
CF17	Number of Neighbors in a community graph	Individual

CF18*	Highest weight among neighbors	Individual
CF19*	Secondary Highest weight among neighbors	Individual
CF20*	Smallest weight among neighbors	Individual

Also, I included features related to users' individual behavior which is related to individual reviews and time. The selected list of features is shown as follows in Table 3.3. For singleton nodes, which do not belong to any communities, I used features in Table 3.3 for classification.

Table 3.3. Features extracted from user's review data. My original features are noted as *.

Number	Feature
IF1	Number of reviews
IF2	Date of First review
IF3	Date of Last review
IF4	Lifetime (IF3-IF2)
IF5	Number of reviews on one day a reviewer wrote the most
IF6	A ratio of reviews on the day a reviewer wrote the most to the total number of reviews (IF5/IF1)
IF7	Entropy of review writing time
IF8	Average time difference of reviews
IF9	Standard Deviation of time difference of reviews

IF10	Ratio of positive (4-5 in 5-scale) reviews
IF11	Ratio of negative (1-2 in 5-scale) reviews
IF12*	Ratio of relatively positive (higher than average) reviews
IF13*	Ratio of relatively negative (lower than average) reviews
IF14	Average Rating
IF15	Standard Deviation of rating

3.6 Reviewer Community Feature Extraction and Spammer Detection

I use random forest method to train the classifier with my extracted features. I compare my proposed method with seven recently developed approaches and a simplified version of my method, which are summarized in Table 3.4.

Table 3.4. Summary of baselines with their methods.

Baselines	Method
GC [95]	Iterative Classification Algorithm
SVM [95]	Support vector machine
SpEaglePlus [81]	Loopy belief propagation
FRAUDAR [69]	Density-based greedy approach
Mzoom [85]	Density-based greedy approach
GSBC [93]	Bi-connected graph-based min cut

DEFRAUDER [63]	Graph embedding
SC-Com(Naïve)	Louvain Method in collusiveness graph
SC-Com	Restricted Louvain Method in collusiveness graph

GC [95] proposed the collective inference problem based on Iterative Classification Algorithm [102]. I compared the result from the result in their paper. They also provided SVM-based approach, whose result is denoted as SVM in Table 3.6.

SpEaglePlus [81] provided a detection algorithm in opinion spam, which utilized loopy belief propagation to reviewer-product network. I excluded parts of linguistic feature parts of SpEaglePlus.

FRAUDAR [69] and Mzoom [85] detects fraudulent blocks from the graph iteratively, so I ran it several times until I get the highest F1-Score using 5-fold cross validation. Also, I constructed same features of my extracted features in fraudulent blocks.

Naïve Louvain is a simpler version of my method, which uses the basic Louvain method for community detection.

GSBC [93] detects spammer groups by dividing the network into bi-connected components and using their spamcity.

DEFRAUDER [63] is a state-of-the-art approach which detects candidate fraud groups and ranking them by mapping reviewers into embedding space.

For a fair comparison, unsupervised methods such as FRAUDAR and Mzoom are converted to supervised methods by extracting the same features which are included in my proposed method in their detected blocks and use random forest method. I called them FRAUDAR+ and Mzoom+. For GSBC, I include my original features into GSBC and name it as GSBC+. For DEFRAUDER, I considered the maximal possible performance from the detected candidate fraud groups to leverage the gap between supervised learning and unsupervised learning.

3.7 Performance Analysis

I set up classifiers using a part of my features, to check the influence of each feature type. First, I perform the classification task using individual features. As community features are defined only on users which belong to any communities, I perform classification separately using them. In the last part, I combine two types of features and perform classification from all data. Table 3.5 includes the precision, recall, and F1 score by the use of each feature.

Table 3.5. Classification performance by varying the scope of features

	Precision	Recall	F1
Individual	0.750	0.804	0.776
Community	0.959	0.947	0.952
Community+Individual	0.940	0.926	0.933

I observe that considering features from each reviewer's local community structure can detect spammers effectively compared with considering individual behavior features. It is clear evidence that SC-Com can separate collusive spammers well from all users. When I consider all features, my result shows a great improvement of precision and recall, compared to the result when using individual features.

Next, I compare the performance of spammer classification of SC-Com with previous baseline approaches in Table 3.6. I included a result of 'Naïve Louvain', which just use original Louvain method, by not restricting the weight using parameter λ in my research.

Table 3.6. Classification performance with baselines

	Precision	Recall	F1-Score
SpEaglePlus [81]	0.936	0.531	0.679
FRAUDAR+ [69]	0.886	0.550	0.680
Mzoom+ [85]	0.914	0.629	0.745
GSBC+ [93]	0.906	0.699	0.789
SVM [95]	0.833	0.827	0.829
DeFrauder [63]	0.941	0.88	0.910
GC [95]	0.939	0.904	0.919
SC-Com (Naïve)	**0.929**	**0.873**	**0.900**
SC-Com	**0.940**	**0.926**	**0.933**

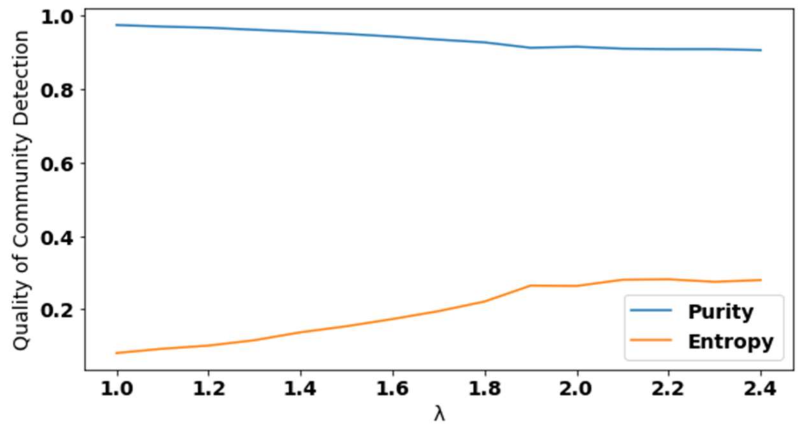
At first, SC-Com performed better than all baselines, especially by showing a higher recall. The main difference between SC-Com and other baselines is significant increase in recall. It means that SC-Com provides high-quality clusters using Restricted Louvain method to extract collusive communities effectively. One reason for such success is the effect of parameter λ, whose effect was shown by the difference between SC-Com(Naïve) and SC-Com. When I instantly apply Louvain methods without filtering the edge weight, there are big communities which contains a large number of both spammers and benign users. By constraining the formation of big communities using λ, my proposed method could find more fine-grained communities.

3.8 Discussion and Implications

In this section, I analyze my result in a practical way, to get insight for opinion spam ecosystem. First, I look at my approach thoroughly by examining the quality of community detection, and the distribution of features among instances. After that, I will investigate the effect of each parameter by setting it in various conditions.

3.8.1 Quality of Community Detection

As I divided reviewer as collusive review groups, I show how my community detection successfully separate spammers and non-spammers primarily. Figure 3.2 shows the quality of community detection based on overall purity and entropy of detected communities as threshold parameter λ varies. It shows that SC-Com successfully discriminates communities of colluding spammers and non-spammers with higher than 90% purity in every parameter condition. Even though the purity and entropy are the best when λ is 1.0, the classification performance shows the best when λ is 1.7. I also show the recall and coverage of community detection method in the second figure in Figure 3.2, as for various λ.



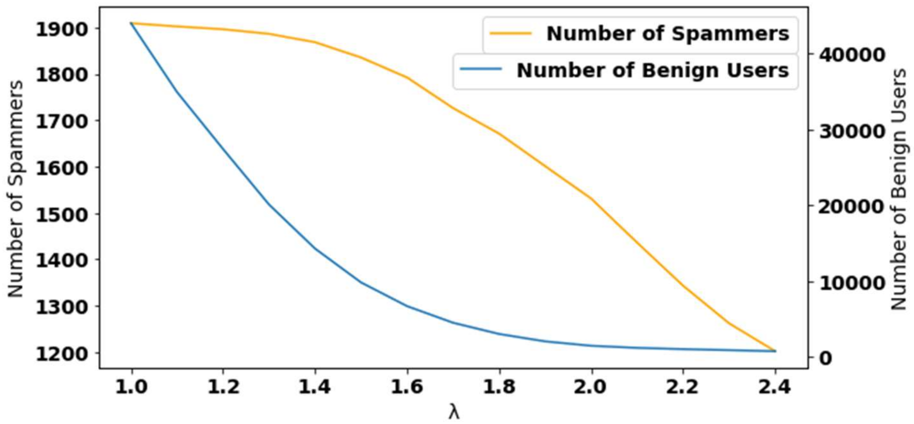


Figure 3.2. The result of community detection. In every λ, the overall purity of communities is higher than 0.9. It means detected communities are mostly spammer groups, or benign groups. Second figure depicts the number of spammers and benign users as λ varies.

3.8.2 Characteristics of Features

Also, I investigate the distribution of each community-based feature as shown in Figure 3.3. I can find that spammers would like to be in a larger community than benign reviewers, and their behavior would be more synchronized due to the distribution of CF4. I detected their collusiveness from calculating the concentration of their behaviors. That is one of the reasons why my proposed method captures collusive spammers effectively. Also, the users in spammer groups tend to write more similar amount of reviews compared to benign users, according to the result of entropy-related values. Furthermore, their reviewing behavior would occur with shorter

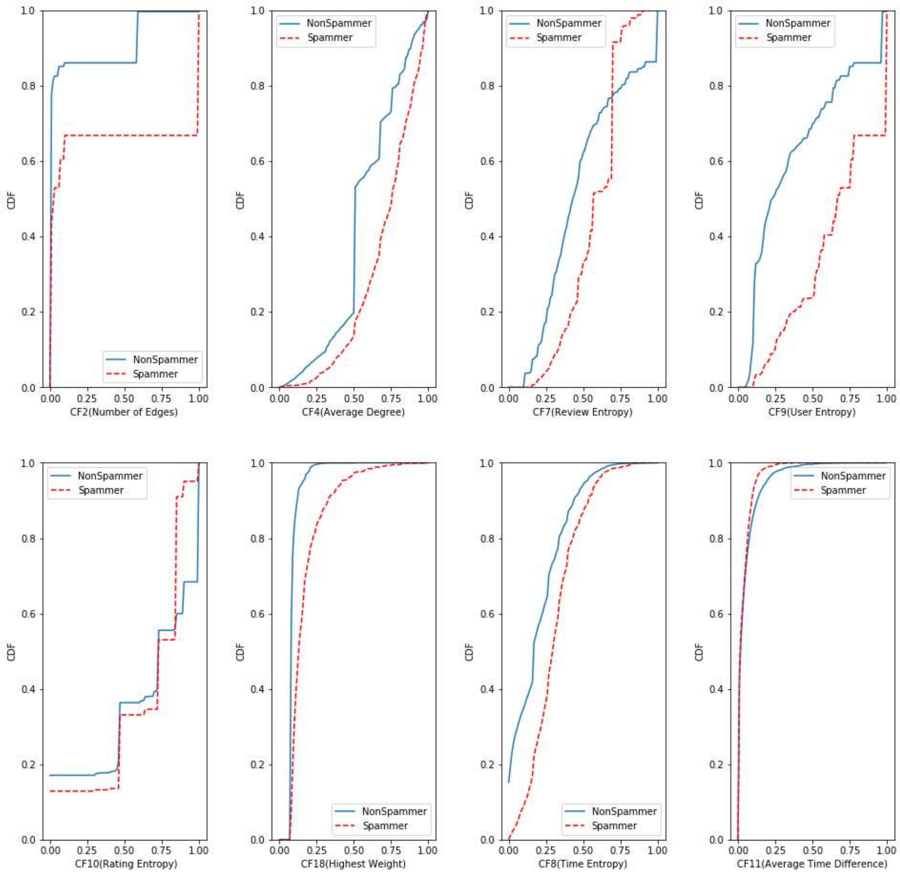


Figure 3.3. Cumulative distribution of community-based feature in all reviewers when λ=1.7.

intervals, which is shown in the result of CF11. CF18 is a notable feature that means spammers have closer neighbors than benign users.

In Figure 3.4, I depicted the distribution of individual behavior features. I could find that there is a difference in the lifetime and the number of reviews between benign users and spammers. A remarkable observation is that the distribution of IF5 and IF6 are different between spammers and non-spammers. Spammers tend to write multiple reviews in a single day, but tho-

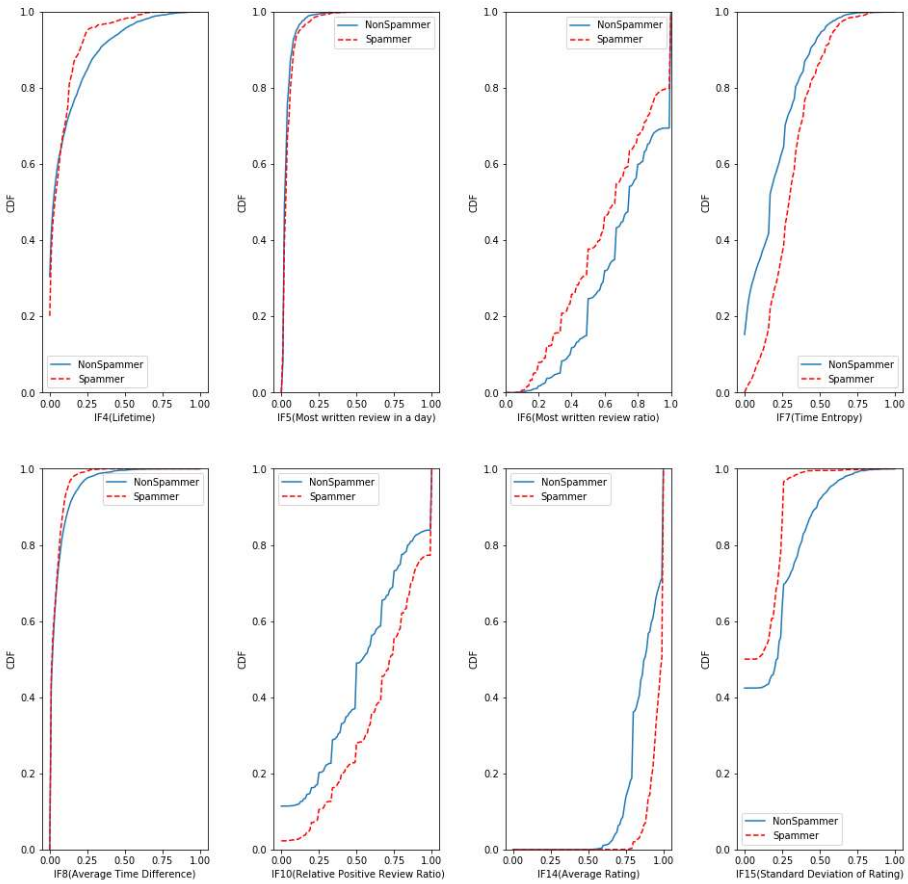


Figure 3.4. Cumulative Distribution of Individual Behavior-based Features

se behavior takes place occasionally, while many benign users tend to write most of their reviews in a single day. Also, the drastic difference in entropy makes this observation clearer.

3.8.3 Robustness of SC-Com

To evaluate the robustness of SC-Com, I set up two scenarios of review spam behaviors and conducted classification tasks: camouflaging and compromised. Those two scenarios are usually considered as potential harm of review spam detection, because spammers hide their identity by adopting those methods. As I assume that these situations occur after the construction of SC-Com, I keep the same parameter of Section 3.4.

The first scenario is based on camouflaging behavior of spammers. As I mention in previous sections, spammers tend to hide their abnormal behaviors by acting as benign reviewers. To model this phenomenon, I randomly choose a portion of spammers and added new benign reviews which are sampled from original benign reviews. Table 3.7. depicts the performance of SC-Com in varying number of camouflaging spammers. In Table 3.7, the f1-score of SC-Com did not drop below 0.919 from 5 accounts to 290 accounts. This result shows that SC-Com is not severely affected by camouflaging behaviors of spammers.

Table 3.7. Classification Performance of SC-Com in Camouflaging Scenario

#Accounts	0	5	10	15	97 (5%)	194 (10%)	290 (15%)
Precision	0.940	0.948	0.947	0.937	0.944	0.934	0.942
Recall	0.926	0.901	0.916	0.940	0.912	0.904	0.898
F1-Score	0.933	0.924	0.932	0.938	0.928	0.919	0.920

Also, I consider compromising accounts as the second scenario, which involves the civil spam attacks by using existing benign accounts to generate spam reviews. Similar to first scenario, I randomly choose a portion of benign users and added new spam reviews which are sampled from reviews of spammers. To make the scenario more reliable, I set up the period of each spamming campaign as 3 weeks and consider all spam reviews' rating as 5. Table 3.8 depicts the performance of SC-Com in varying number of compromised accounts. In Table 3.8. the f1-score of SC-Com is not dropped below 0.926. Those means that almost more than 80% of compromised accounts can be quickly detected as spammers using SC-Com.

Table 3.8. Classification Performance of SC-Com in Compromised Accounts Scenario

#Accounts	0	5	10	15	97 (5%)	194 (10%)	290 (15%)
Precision	0.940	0.943	0.949	0.937	0.936	0.937	0.933
Recall	0.926	0.910	0.918	0.929	0.917	0.914	0.924
F1-Score	0.933	0.926	0.933	0.933	0.927	0.926	0.928

3.8.4 Parameter Study

As I set up the threshold parameter λ as restrictions for community detection, I show the impact of threshold parameter λ by doing whole framework with varying values. With a high value of λ, communities are formed with smaller and sparser sizes. As shown in Figure 3.5, my framework is robust for varying λ, so it can be tuned for various domains with different atmosphere.

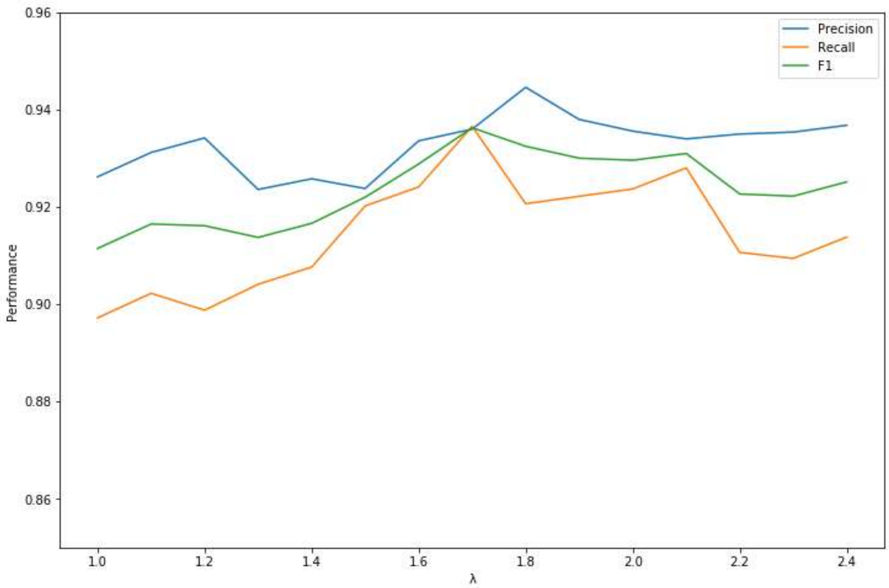


Figure 3.5. Parameter Sensitivity of λ. As my framework shows the best performance when λ=1.7 with the highest F1, the overall performance remains stable when λ varies.

3.9 Summary

In summary, I performed a novel opinion spam detection using community-based anomalies. In the previous work, using empirical detection algorithm such as text analysis were shown not to be useful as the spammers become more intelligent and disguise their writing styles. To detect those spammers that are getting hard to be found, I focus on the collusiveness of their behavior, which is closely related to the purpose of opinion spam writing. By catching anomalies in two sides, one is community-based and one is individual behavior-based, I introduce an effective way to detect collusive spammers by dividing users into smaller communities based on their behavioral collusiveness. From user communities, I extracted features related to their activeness and anomalies, and from individual behavior, I extracted time-based features and rating-based features. Also, I proposed a supervised classification method using features related to community behavior and individual behavior. From the analysis using a real-world dataset, my method outperformed previous state-of-the-art methods especially by showing a higher recall. From the analysis of parameter selection, I showed that my proposed scheme could be used flexibly in other domains, by not requiring linguistic data and providing learnable parameters. My research shows the importance of considering collusiveness in review spam detection, by showing significant improvements in a real-world dataset.

While I utilized some useful features from the constructed community graph, there would be more information in the graph itself. I would expect the improvement if I consider more detailed approaches such as network embedding in a deep neural network. Also, my research on detecting collusiveness can be used to solve various problems which are getting concerns in social media, such as conflicts in online communities, cyberbullying on the social network service, and so forth.

Chapter 4 Conclusion

In this paper, I proposed ANES and SC-Com, which utilizes and detect social relation of users in online platforms. ANES performs representation learning on User-POI check-in data for unsupervised social link inference. I design the aspect-oriented relationship between users and POIs to learn both the pairwise relation and network structures. From extensive experiments, I show ANES is superior to other state-of-the-art methods by showing the result of the unsupervised link prediction task.

SC-Com extracted features from users review data related to their activeness and anomalies. From individual behavior, I extracted time-based features and rating-based features. After that, I proposed a supervised classification method using features related to community behavior and individual behavior. From the analysis using a real-world dataset, SC-Com outperformed previous state-of-the-art methods especially by showing a higher recall.

Bibliography

[1] Backes, M., Humbert, M., Pang, J., Zhang, Y., 2017. walk2friends: Inferring social links from mobility profiles, in: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pp. 1943–1957.

[2] Bao, J., Zheng, Y., Mokbel, M.F., 2012. Location-based and preference aware recommendation using sparse geo-social networking data, in: Proceedings of the 20th international conference on advances in geographic information systems, pp. 199–208.

[3] Barabasi, A.L., Albert, R., 1999. Emergence of scaling in random networks. science 286, 509–512.

[4] Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O., 2013. Translating embeddings for modeling multi-relational data, in: Advances in neural information processing systems, pp. 2787–2795.

[5] Byun, H., Kim, C.K., 2019. When friends move: A deep learning-based approach for friendship prediction in mobility network (poster), in: Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services, pp. 520–521.

[6] Cai, L., Li, J., Wang, J., Ji, S., 2021. Line graph neural networks for link prediction. IEEE Transactions on Pattern Analysis and Machine Intelligence

[7] Chung, J., Gulcehre, C., Cho, K., Bengio, Y., 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 .

[8] Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S., 2018. Convolutional 2d knowledge graph embeddings, in: Proceedings of the AAAI Conference on Artificial Intelligence.

[9] Dong, Y., Chawla, N.V., Swami, A., 2017. metapath2vec: Scalable representation learning for heterogeneous networks, in: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, pp. 135–144.

[10] Fan, W., Ma, Y., Li, Q., He, Y., Zhao, E., Tang, J., Yin, D., 2019. Graph neural networks for social recommendation, in: The World Wide Web Conference, pp. 417–426.

[11] Feng, M.H., Hsu, C.C., Li, C.T., Yeh, M.Y., Lin, S.D., 2019. Marine: Multi-relational network embeddings with relational proximity and node attributes, in: The World Wide Web Conference, pp. 470–479.

[12] Gao, H., Tang, J., Liu, H., 2012. Exploring social-historical ties on location-based social networks, in: Sixth International AAAI conference on weblogs and social media.

[13] Gao, Q., Trajcevski, G., Zhou, F., Zhang, K., Zhong, T., Zhang, F., 2018. Trajectory-based social circle inference, in: Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 369–378.

[14] Gao, Q., Zhou, F., Zhang, K., Trajcevski, G., Luo, X., Zhang, F., 2017. Identifying human mobility via trajectory embeddings., in: IJCAI, pp. 1689–1695.

[15] Gilmer, J., Schoenholz, S.S., Riley, P.F., Vinyals, O., Dahl, G.E., 2017. Neural message passing for quantum chemistry, in: International Conference on Machine Learning, PMLR. pp. 1263–1272.

[16] Gong, L., Cheng, Q., 2019. Exploiting edge features for graph neural networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9211–9219.

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

[18] Hang, M., Pytlarz, I., Neville, J., 2018. Exploring student check-in behavior for improved point-of-interest prediction, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, ACM. pp. 321–330.

[19] He, C., Peng, C., Li, N., Chen, X., Guo, L., 2018. Exploiting spatiotemporal features to infer friendship in location-based social networks, in: Pacific Rim International Conference on Artificial Intelligence, Springer. pp. 395–403.

[20] Hong, H., Guo, H., Lin, Y., Yang, X., Li, Z., Ye, J., 2020. An attention-based graph neural network for heterogeneous structural learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 4132–4139.

[21] Huang, S., Ma, Q., Yang, C., Yao, Y., 2021. Link prediction with multiple structural attentions in multiplex networks, in: 2021 International Joint Conference on Neural Networks (IJCNN), pp. 1–9. doi:10.1109/IJCNN52387.2021.9533609.

[22] Huo, Y., Chen, B., Tang, J., Zeng, Y., 2021. Privacy-preserving pointof-interest recommendation based on geographical and social influence. Information Sciences 543, 202–218.

[23] Huynh, T.T., Tong, V.V., Nguyen, T.T., Jo, J., Yin, H., Nguyen, Q.V.H., 2022. Learning holistic interactions in lbsns with high-order, dynamic, and multi-role contexts. IEEE Transactions on Knowledge and Data Engineering, 1–1doi:10.1109/TKDE.2022.3150792.

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

[25] Li, J., Zeng, F., Xiao, Z., Jiang, H., Zheng, Z., Liu, W., Ren, J., 2020. Drive2friends: Inferring social relationships from individual vehicle mobility data. IEEE Internet of Things Journal.

[26] Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X., 2015. Learning entity and relation embeddings for knowledge graph completion, in: Twenty-ninth AAAI conference on artificial intelligence.

[27] Liu, N., Tan, Q., Li, Y., Yang, H., Zhou, J., Hu, X., 2019. Is a single vector enough? exploring node polysemy for network embedding, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 932–940.

[28] Liu, Q., Wu, S., Wang, L., Tan, T., 2016. Predicting the next location: A recurrent model with spatial and temporal contexts, in: Proceedings of the AAAI Conference on Artificial Intelligence.

[29] Maaten, L.v.d., Hinton, G., 2008. Visualizing data using t-sne. Journal of machine learning research 9, 2579–2605.

[30] Mukherjee, S., G¨unnemann, S., 2019. Ghostlink: Latent network inference for influence-aware recommendation, in: The World Wide Web

Conference, pp. 1310–1320.

[31] Nathani, D., Chauhan, J., Sharma, C., Kaul, M., 2019. Learning attention-based embeddings for relation prediction in knowledge graphs. arXiv preprint arXiv:1906.01195 .

[32] Perozzi, B., Al-Rfou, R., Skiena, S., 2014. Deepwalk: Online learning of social representations, in: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 701–710.

[33] Pham, H., Shahabi, C., Liu, Y., 2016. Inferring social strength from spatiotemporal data. ACM Transactions on Database Systems (TODS) 41, 1–47.

[34] Qian, T., Liu, B., Nguyen, Q.V.H., Yin, H., 2019. Spatiotemporal representation learning for translation-based poi recommendation. ACM Transactions on Information Systems (TOIS) 37, 1–24.

[35] Schlichtkrull, M., Kipf, T.N., Bloem, P., Van Den Berg, R., Titov, I., Welling, M., 2018. Modeling relational data with graph convolutional networks, in: European semantic web conference, Springer. pp. 593–607.

[36] Shi, Y., Gui, H., Zhu, Q., Kaplan, L., Han, J., 2018. Aspem: Embedding learning by aspects in heterogeneous information networks, in: Proceedings

of the 2018 SIAM International Conference on Data Mining, SIAM. pp. 144–152.

[37] Song, C., Wang, B., Jiang, Q., Zhang, Y., He, R., Hou, Y., 2021. Social recommendation with implicit social influence, in: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1788–1792.

[38] Su, Y., Li, X., Tang, W., Xiang, J., He, Y., 2018. Next check-in location prediction via footprints and friendship on location-based social networks, in: 2018 19th IEEE International Conference on Mobile Data Management (MDM), IEEE. pp. 251–256.

[39] Tang, J., Qu, M., Mei, Q., 2015a. Pte: Predictive text embedding through large-scale heterogeneous text networks, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1165–1174.

[40] Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q., 2015b. Line: Large-scale information network embedding, in: Proceedings of the 24th international conference on world wide web, pp. 1067–1077.

[41] Veliˇckovi´c, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y., 2017. Graph attention networks. arXiv preprint arXiv:1710.10903 .

[42] Wang, H., Meng, Q., Fan, J., Li, Y., Cui, L., Zhao, X., Peng, C., Chen, G., Du, X., 2020. Social influence does matter: User action prediction for in-feed advertising, in: Proceedings of the AAAI Conference on artificial intelligence, pp. 246–253.

[43] Wang, Z., Zhang, J., Feng, J., Chen, Z., 2014. Knowledge graph embedding by translating on hyperplanes, in: Twenty-Eighth AAAI conference on artificial intelligence.

[44] Wu, Y., Lian, D., Jin, S., Chen, E., 2019. Graph convolutional networks on user mobility heterogeneous graphs for social relationship inference., in: IJCAI, pp. 3898–3904.

[45] Xie, M., Yin, H., Wang, H., Xu, F., Chen, W., Wang, S., 2016. Learning graph-based poi embedding for location-based recommendation, in: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, ACM. pp. 15–24.

[46] Xu, S., Pi, D., Cao, J., Fu, X., 2021. Hierarchical temporal–spatial preference modeling for user consumption location prediction in geosocial networks. Information Processing & Management 58, 102715.

[47] Xu, W., Chow, C.Y., Zhang, J.D., 2013. Calba: capacity-aware location-based advertising in temporary social networks, in: Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 364–373.

[48] Yang, D., Qu, B., Yang, J., Cudre-Mauroux, P., 2019. Revisiting user mobility and social relationships in lbsns: A hypergraph embedding approach, in: The world wide web conference, pp. 2147–2157.

[49] Yang, D., Qu, B., Yang, J., Cudr´e-Mauroux, P., 2020. Lbsn2vec++: Heterogeneous hypergraph embedding for location-based social networks. IEEE Transactions on Knowledge and Data Engineering .

[50] Yin, H., Zhou, X., Shao, Y., Wang, H., Sadiq, S., 2015. Joint modeling of user check-in behaviors for point-of-interest recommendation, in: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, ACM. pp. 1631–1640.

[51] Yu, F., Cui, L., Guo, W., Lu, X., Li, Q., Lu, H., 2020. A category-aware deep model for successive poi recommendation on sparse check-in data, in: Proceedings of The Web Conference 2020, pp. 1264–1274.

[52] Yu, Y., Wang, H., Li, Z., 2018. Inferring mobility relationship via graph embedding. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 2, 1–21.

[53] Zhang, W., Lai, X., Wang, J., 2020. Social link inference via multiview matching network from spatiotemporal trajectories. IEEE transactions on neural networks and learning systems .

[54] Zhao, K., Bai, T., Wu, B., Wang, B., Zhang, Y., Yang, Y., Nie, J.Y., 2020. Deep adversarial completion for sparse heterogeneous information network embedding, in: Proceedings of The Web Conference 2020, pp. 508–518

[55] Zheng, J., Qin, Z., Wang, S., Li, D., 2022. Attention-based explainable friend link prediction with heterogeneous context information. Information Sciences 597, 211–229. doi:https://doi.org/10.1016/j.ins.2022.03.010.

[56] Zhou, F., Gao, Q., Trajcevski, G., Zhang, K., Zhong, T., Zhang, F., 2018. Trajectory-user linking via variational autoencoder., in: IJCAI, pp. 3212–3218.

[57] Huang, C., Xu, H., Xu, Y., Dai, P., Xia, L., Lu, M., Bo, L., Xing, H., Lai, X., & Ye, Y. (2021). Knowledge-aware Coupled Graph Neural Network for

Social Recommendation. Proceedings of the AAAI Conference on Artificial Intelligence, 35(5), 4115-4122. https://doi.org/10.1609/aaai.v35i5.16533

[58] M. McPherson, L. Smith-Lovin, and J. M. Cook. Birds of a feather: Homophily in social networks. Annual review of sociology, 27(1):415–444, 2001.

[59] Akoglu, L., Tong, H., & Koutra, D. (2015). Graph based anomaly detection and description: a survey. Data mining and knowledge discovery, 29(3), 626-688.

[60] Blondel, V. D., Guillaume, J. L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. Journal of statistical mechanics: theory and experiment, 2008(10), P10008.

[61] Celik, M., & Dokuz, A. S. (2018). Discovering socially similar users in social media datasets based on their socially important locations. Information Processing & Management, 54(6), 1154-1168.

[62] Choo, E., Yu, T., & Chi, M. (2015, July). Detecting opinion spammer groups through community discovery and sentiment analysis. In IFIP annual conference on data and applications security and privacy (pp. 170-187). Springer, Cham.

[63] Dhawan, S., Gangireddy, S. C. R., Kumar, S., & Chakraborty, T. (2019, August). Spotting collective behaviour of online frauds in customer reviews. In Proceedings of the 28th International Joint Conference on Artificial Intelligence (pp. 245-251). AAAI Press.

[64] Dong, L. Y., Ji, S. J., Zhang, C. J., Zhang, Q., Chiu, D. W., Qiu, L. Q., et al. (2018). An unsupervised topic-sentiment joint probabilistic model for detecting deceptive reviews. Expert Systems with Applications, 114, 210–223.

[65] El-Mawass, N., Honeine, P., & Vercouter, L. (2020). SimilCatch: Enhanced social spammers detection on Twitter using Markov Random Fields. Information Processing & Management, 57(6), 102317.

[66] Fei, G., Mukherjee, A., Liu, B., Hsu, M., Castellanos, M., & Ghosh, R. (2013). Exploiting burstiness in reviews for review spammer detection. Icwsm, 13, 175-184.

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

[68] Hooi, B., Shah, N., Beutel, A., Günnemann, S., Akoglu, L., Kumar, M., ... & Faloutsos, C. (2016, June). Birdnest: Bayesian inference for ratings-fraud detection. In Proceedings of the 2016 SIAM International Conference on Data Mining (pp. 495-503). Society for Industrial and Applied Mathematics.

[69] Hooi, B., Song, H. A., Beutel, A., Shah, N., Shin, K., & Faloutsos, C. (2016, August). Fraudar: Bounding graph fraud in the face of camouflage. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 895-904).

[70] Hu, N., Liu, L., & Zhang, J. J. (2008). Do online reviews affect product sales? The role of reviewer characteristics and temporal effects. Information Technology and management, 9(3), 201-214.

[71] Hussain, N., Turab Mirza, H., Rasool, G., Hussain, I., & Kaleem, M. (2019). Spam review detection techniques: A systematic literature review. Applied Sciences, 9(5), 987.

[72] Jindal, N., & Liu, B. (2007, May). Review spam detection. In Proceedings of the 16th international conference on World Wide Web (pp. 1189-1190).

[73] Jindal, N., & Liu, B. (2008, February). Opinion spam and analysis. In Proceedings of the 2008 international conference on web search and data mining (pp. 219-230).

[74] Kumar, D., Shaalan, Y., Zhang, X., & Chan, J. (2018, June). Identifying singleton spammers via spammer group detection. In Pacific-Asia Conference on Knowledge Discovery and Data Mining (pp. 656-667). Springer, Cham.

[75] Kumar, S., Hooi, B., Makhija, D., Kumar, M., Faloutsos, C., & Subrahmanian, V. S. (2018, February). Rev2: Fraudulent user prediction in rating platforms. In Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining (pp. 333-341).

[76] Li, H., Fei, G., Wang, S., Liu, B., Shao, W., Mukherjee, A., & Shao, J. (2017, April). Bimodal distribution and co-bursting in review spam detection. In Proceedings of the 26th International Conference on World Wide Web (pp. 1063-1072).

[77] Liu, Y., Pang, B., & Wang, X. (2019). Opinion spam detection by incorporating multimodal embedded representation into a probabilistic review graph. Neurocomputing, 366, 276-283.

[78] Mukherjee, A., Kumar, A., Liu, B., Wang, J., Hsu, M., Castellanos, M., & Ghosh, R. (2013, August). Spotting opinion spammers using behavioral footprints. In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 632-640).

[79] Noekhah, S., binti Salim, N., & Zakaria, N. H. (2020). Opinion spam detection: Using multi-iterative graph-based model. Information Processing & Management, 57(1), 102140.

[80] Papadopoulos, S., Kompatsiaris, Y., Vakali, A., & Spyridonos, P. (2012). Community detection in social media. Data Mining and Knowledge Discovery, 24(3), 515-554.

[81] Rayana, S., & Akoglu, L. (2015, August). Collective opinion spam detection: Bridging review networks and metadata. In Proceedings of the 21th acm sigkdd international conference on knowledge discovery and data mining (pp. 985-994).

[82] Ren, Y., Ji, D., & Zhang, H. (2014, October). Positive unlabeled learning for deceptive reviews detection. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 488-498).

[83] Savage, D., Zhang, X., Yu, X., Chou, P., & Wang, Q. (2015). Detection of opinion spam based on anomalous rating deviation. Expert Systems with Applications, 42(22), 8650-8657.

[84] Shehnepoor, S., Salehi, M., Farahbakhsh, R., & Crespi, N. (2017). Netspam: A network-based spam detection framework for reviews in online social media. IEEE Transactions on Information Forensics and Security, 12(7), 1585-1595.

[85] Shin, K., Hooi, B., & Faloutsos, C. (2016, September). M-zoom: Fast dense-block detection in tensors with quality guarantees. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases (pp. 264-280). Springer, Cham.

[86] Shin, K., Hooi, B., Kim, J., & Faloutsos, C. (2017, February). D-cube: Dense-block detection in terabyte-scale tensors. In Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (pp. 681-689).

[87] Stanton, G., & Irissappane, A. A. (2019). GANs for semi-supervised opinion spam detection. Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, Main track. Pages 5204-5210.

[88] Sun, Y., & Loparo, K. (2019, November). Opinion spam detection based on heterogeneous information network. In 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI) (pp. 1156-1163). IEEE.

[89] Tian, Y., Mirzabagheri, M., Tirandazi, P., & Bamakan, S. M. H. (2020). A non-convex semi-supervised approach to opinion spam detection by ramp-one class SVM. Information Processing & Management, 102381.

[90] Wang, G., Wilson, C., Zhao, X., Zhu, Y., Mohanlal, M., Zheng, H., & Zhao, B. Y. (2012, April). Serf and turf: crowdturfing for fun and profit. In Proceedings of the 21st international conference on World Wide Web (pp. 679-688).

[91] Wang, G., Xie, S., Liu, B., & Philip, S. Y. (2011, December). Review graph based online store review spammer detection. In 2011 IEEE 11th International Conference on Data Mining (pp. 1242-1247). IEEE.

[92] Wang, X., Liu, K., He, S., & Zhao, J. (2016, November). Learning to represent review with tensor decomposition for spam detection. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (pp. 866-875).

[93] Wang, Z., Gu, S., Zhao, X., & Xu, X. (2018). Graph-based review spammer group detection. Knowledge and Information Systems, 55(3), 571-597.

[94] Wang, Z., Hou, T., Song, D., Li, Z., & Kong, T. (2016). Detecting review spammer groups via bipartite graph projection. The Computer Journal, 59(6), 861-874.

[95] Xu, C., Zhang, J., Chang, K., & Long, C. (2013, October). Uncovering collusive spammers in Chinese review websites. In Proceedings of the 22nd ACM international conference on Information & Knowledge Management (pp. 979-988)..

[96] Ye, J., & Akoglu, L. (2015, September). Discovering opinion spammer groups by network footprints. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases (pp. 267-282). Springer, Cham.

[97] Ye, J., Kumar, S., & Akoglu, L. (2016). Temporal opinion spam detection by multivariate indicative signals. In ICWSM.

[98] Zhang, W., Du, Y., Yoshida, T., & Wang, Q. (2018). DRI-RCNN: An approach to deceptive review identification using recurrent convolutional neural network. Information Processing & Management, 54(4), 576-592.

[99] Zheng, H., Xue, M., Lu, H., Hao, S., Zhu, H., Liang, X., & Ross, K. (2018). Smoke screener or straight shooter: Detecting elite sybil attacks in user-review social networks. In NDSS.

[100] Ka-Wei Roy Lee and Ee-Peng Lim. 2016. Friendship Maintenance and Prediction in Multiple Social Networks. In Proceedings of the 27th ACM Conference on Hypertext and Social Media (HT '16). Association for Computing Machinery, New York, NY, USA, 83–92. https://doi.org/10.1145/2914586.2914593

[101] Jiang, M., Cui, P., Beutel, A., Faloutsos, C., & Yang, S. (2014, August). Catchsync: catching synchronized behavior in large directed graphs. In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 941-950).

[102] Neville, Jennifer, and David Jensen. "Iterative classification in relational data." Proc. AAAI-2000 workshop on learning statistical models from relational data. 2000.

초 록

웹 기반 서비스의 폭발적인 발달로 사용자들은 온라인 상에서 폭넓게 연결되고 있다. 온라인 플랫폼 상에서, 사용자들은 서로에게 영향을 주고받으며 의사 결정에 그들의 경험과 의견을 반영하는 경향을 보인다. 본 학위 논문에서는 대표적인 온라인 플랫폼인 소셜 네트워크 서비스와 이커머스 플랫폼에서의 사용자 행동에 대해 연구하였다.

온라인 플랫폼에서의 사용자 행동은 사용자와 플랫폼 구성 요소 간의 관계로 표현할 수 있다. 사용자의 구매는 사용자와 상품 간의 관계로, 사용자의 체크인은 사용자와 장소 간의 관계로 나타내진다. 여기에 행동의 시간과 레이팅, 태그 등의 정보가 포함될 수 있다.

본 연구에서는 두 플랫폼에서 정의된 사용자의 행동 그래프에 영향을 미치는 잠재 네트워크를 파악하는 연구를 제시한다. 위치 기반의 소셜 네트워크 서비스의 경우 특정 장소에 방문하는 체크인 형식으로 많은 포스트가 만들어지는데, 사용자의 장소 방문은 사용자 간에 사전에 존재하는 친구 관계에 의해 영향을 크게 받는다. 사용자 활동 네트워크의 저변에 잠재된 사용자 간의 관계를 파악하는 것은 활동 예측에 도움이 될 수 있으며, 이를 위해 본 논문에서는 비지도학습 기반으로 활동 네트워크로부터 사용자 간 사회적 관계를 추출하는 연구를 제안하였다.

기존에 연구되었던 방법들은 두 사용자가 동시에 방문하는 행위인 co-visitation 을 중점적으로 고려하여 사용자 간의 관계를 예측하거나,

네트워크 임베딩 또는 그래프 신경망(GNN)을 사용하여 표현 학습을 수행하였다. 그러나 이러한 접근 방식은 주기적인 방문이나 장거리 이동 등으로 대표되는 사용자의 행동 패턴을 잘 포착하지 못한다. 행동 패턴을 더 잘 학습하기 위해, ANES 는 사용자 컨텍스트 내에서 사용자와 관심 지점(POI) 간의 측면(Aspect) 지향 관계를 학습한다. ANES 는 User-POI 이분 그래프의 구조에서 사용자의 행동을 여러 개의 측면으로 나누고, 각각의 관계를 고려하여 행동 패턴을 추출하는 최초의 비지도학습 기반 접근 방식이다. 실제 LBSN 데이터에서 수행된 광범위한 실험에서, ANES 는 기존에 제안되었던 기법들보다 높은 성능을 보여준다.

위치 기반 소셜 네트워크와는 다르게, 이커머스의 리뷰 시스템에서는 사용자들이 능동적인 팔로우/팔로잉 등의 행위를 수행하지 않고도 플랫폼에 의해 서로의 정보를 주고받고 영향력을 행사하게 된다. 이와 같은 사용자들의 행동 특성은 리뷰 스팸에 의해 쉽게 악용될 수 있다. 리뷰 스팸은 실제 사용자의 의견을 숨기고 평점을 조작하여 잘못된 정보를 전달하는 방식으로 이루어진다. 나는 이를 해결하기 위해 사용자 리뷰 데이터에서 사용자 간 사전 공모성(Collusiveness)의 가능성을 찾고, 이를 스팸 탐지에 활용한 방법인 SC-Com 을 제안한다. SC-Com 은 행동의 공모성으로부터 사용자 간 공모 점수를 계산하고 해당 점수를 바탕으로 전체 사용자를 유사한 사용자들의 커뮤니티로 분류한다. 그 후 스팸 유저와 일반 유저를 구별하는 데에 중요한 그래프 기반의 특징을 추출하여 감독 학습 기반의 분류기의 입력 데이터로 활용하는 방법을 제시한다. SC-Com 은 공모성을 갖는 스팸 유저의 집합을

효과적으로 탐지한다. 실제 데이터셋을 이용한 실험에서, SC-Com 은 기존 논문들 대비 스팸 탐지에 뛰어난 성능을 보여주었다.

위 논문에서 다양한 데이터에 대해 연구된 암시적 연결망 탐지 모델은 레이블이 없는 데이터에 대해서도 사전에 연결되었을 가능성이 높은 사용자들을 예측하므로, 실시간 위치 데이터나, 앱 사용 데이터 등의 다양한 데이터에서 활용할 수 있는 유용한 정보를 제공하여 광고 추천 시스템이나, 악성 유저 탐지 등의 분야에서 기여할 수 있을 것으로 기대한다.

주요어: 소셜 네트워크 분석, 스팸 탐지, 그래프 학습, 사회적 링크 예측

학 번: 2016-21210