Competitive Relationship Prediction for Points of Interest: A Neural Graphlet Based Approach

Jingbo Zhou, Tao Huang, Shuangli Li, Renjun Hu, Yanchi Liu,
Yanjie Fu, Member, IEEE, Hui Xiong, Fellow, IEEE

Abstract—Competition between Points of Interest (POIs) refers to the situation in which two POIs directly or indirectly provide similar services to secure businesses. A large portion of prior studies on competition analysis focuses on mining textual data, e.g., news articles and social comments. However, the increasing availability of human mobility and mobile query data enables a new paradigm for analyzing the competitive relationships among POIs, which remains largely unexplored. To this end, in this paper, we attempt to mine large-scale online map search query data for better understanding POI competitive relationships. Based on a co-query POI graph built from the map search query data, we develop a novel neural graphlet-based prediction framework to predict the competitive relationships among POIs. A unique perspective of our model is to infer latent POI competitive relationships by integrating multiple distinct factors, e.g., graphlet structure, geographical distance, and regional features, reflected in map search query data and POI data. Finally, we conduct extensive experiments on real-world datasets to demonstrate the effectiveness of the proposed framework, and show that our framework outperforms all baselines with a significant margin in all evaluation metrics.

Index Terms—Point of Interest, Competitive Relationship Prediction, Graphlet

1 Introduction

A competitive relationship between two Points of Interest (POIs), especially those providing similar services directly or indirectly, refers to the degree of their competition in securing businesses from a third party in an urban area. A POI is a specific point location that someone may find useful services. Different popular categories of POIs in an urban area include bars, retail stores, and restaurants, etc. Though city daily human activities take place at POIs in the urban area, POIs usually have to strive for limited users to survive, which enhances the provision of high-quality and beneficial services for users in an urban area.

Understanding the competitive environment of POIs is important, both on the individual POI level and the city level. On the individual POI level, being aware of the commercially competitive degree between the POI and its competitors is vital for the business owner to maintain the prosperity of the POI business. For example, the business owner can analyze competitors' pricing strategy, design a special marketing campaign to attract more customers, and make appropriate response strategies for competitive incursions from competitors. On the city level, the study of competitive relationships among POIs is essential for government and regional administrators to understand the regional competitive environment [1], [2] and to make sustainable regional planning, aiming to enhance the urban vibrancy and optimize the urban ecosystem.

- *Jingbo Zhou, Tao Huang, Shuangli Li, Renjun Hu are with the Baidu Research, Business Intelligence Lab. E-mail: {zhoujingbo, huangtao40, lishuangli, hurenjun01}@baidu.com*
- *Yanchi Liu is with the Rutgers University. E-mail: yanchi.liu@rutgers.edu*
- *Yanjie Fu is with the University of Central Florida. E-mail: yanjie.fu@ucf.edu*
- *Hui Xiong is with the Rutgers University. E-mail: hxiong@rutgers.edu*

J. Zhou and H. Xiong are the corresponding authors.

Existing studies on competitive relationship prediction usually consider identifying entities that are compared in the texts of news articles, social networks, and web pages [3], [4], [5]. However, these text mining-based approaches might have difficulty in identifying competitive relationships for POIs since the comparative evidence for POIs is often absent in text data [4]. For example, it may be possible to find comparative patterns for the brands (e.g. "KFC v.s. Mcdonalds"), but it is quite rare to find the specific name of a POI (e.g. a KFC store) in a sentence.

An alternative approach to mine competitive relationships among POIs is to analyze the user check-in data. Intuitively, if two POIs have similar businesses and a portion of overlapped customers, they are very likely to have a competitive relationship. However, user check-in data is extremely sparse in practice, e.g., the sparsity of a Gowalla check-in dataset is around 99.98% [6]. Therefore, even we can mine a few of competitive relationships from text data and check-in data, it is still quite desirable to build an advanced machine learning model upon the mined relationships to predict more competitive relationships from map query data.

To this end, in this paper, we conduct a novel study of identifying competitive relationships among POIs through analyzing large-scale user behavior data – online map search query data. Our work provides a new data-driven research paradigm for the task. The map search query data records users' actions (e.g., search, click, and view) on POIs at online map service platforms, such as Google Maps and Baidu Maps. Compared with the traditional text and check-in data, the map search query data is much more plentiful and broadly applicable to a whole bunch of POIs. To the best of our knowledge, we are among the first to investigate POI competitive relationship prediction based on map search query data.

The discovered competitive relationships from text or check-in data could also be used as seed relationships or ground truth for training our model. However, the text data and check-in data may be sparse for POIs. For example, many POIs may have little text data (e.g. reviews) which can be used to extract competitive relationships. But the map search query data is much plentiful for POIs which provide services. From this perspective, the tasks of competitive relationship prediction from text data or check-in data can be orthogonal, and yet complementary to our problem.

However, identifying POI competitive relationships from map search query data is challenging. Observe that, for a pair of POIs that have been searched by many common users, there exist two opposite facts: 1) the two POIs are competitive and people are making a choice between them; and 2) the two POIs are complementary (instead of competitive) and people are planning a route to visit both of them. If the two co-searched POIs are of the same category, someone may argue there is a competitive relationship as long as the co-search frequency exceeds a threshold. However, we note that it is almost impossible to define these handcrafted rules since the thresholds are diverse owing to various factors such as the density of users, distribution of POIs, and user preference in different regions (e.g., downtown area vs. suburban district). For example, the threshold should be high at the city center, and be low at a suburban district. We should set many different thresholds for them. Hence, it is hard to simply define some handcrafted filter rules on the co-query POI graph to mine competitive relationships directly. If manually-intensively setting adaptive thresholds, it is also easy to fall in a local minimum. Our experiments in Section 5.3 also demonstrate that such heuristic methods and simple classification methods cannot work well. Therefore, how to build a model to capture useful signals (but some signals are not corrected) from map search query data is a unique research challenge.

We thus propose a novel Neural Graphlet-based Prediction framework (named NGP) to identify competitive POI relationships from map search query data. An illustration of the overview of NGP is demonstrated in Figure 1. We first build a co-query POI graph based on the map search query data, whose nodes are POIs and edge weights denote the numbers of times that users search both POIs within a short time interval. Then we extract frequent graphlets from the co-query POI graph by a random walk algorithm. Finally, upon the mined graphlets, we devise a novel Neural Graphlet-based Relationship Prediction model (we refer to it as NGRP model) which has a transformer-like architecture with self attention and cross attention mechanism. NGRP model has a set of carefully designed Graphlet Learning Blocks (GLBs) to process graphlets. Besides, other important factors like region features, the distance between POIs, and count features based on frequent graphlets are also investigated in our framework. We postpone a brief explanation of the work process of the NGP framework in Section 2.2.

Our framework has several novel perspectives. First, it can better capture the context information for for a POI pair in the co-query POI graph to better reveal the competitive relationships. To achieve a good prediction performance of NGP, we need to get a good description "signature" between two POIs, whereas the graphlet can better exploit the topological structures of the co-query POI graph. Second, its transformer-like architecture with self attention and cross attention can fully exploit the latent dependency among POIs, regions and the set of mined graphlets. Third, it takes both the geographical and regional features into consideration for prediction. These unique advantages of the NGP framework result in a significant improvement in the prediction performance compared with baselines. According to our extensive evaluation on two real-life datasets, we find that our model can consistently outperform many baselines with a large margin in all evaluation metrics. Each component of the framework is also verified to be effective in general.

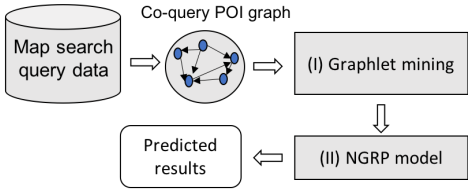


Fig. 1. An overview of NGP framework.

The contributions of this paper can be summarized as:

- We are the first to exploit the large-scale online map search query data to to study POI competitive relationship prediction, which is a complementary of competitive relationship prediction by other data sources (text or check-in data).
- We propose a novel NGP framework to predict the competitive relationship. The core of our framework is a prediction model that has a transformer-like structure with self attention and cross attention mechanism to utilize the features extracted from graphlets, POIs, and regions.
- We conducted an extensive evaluation of the NGP framework on two real-life datasets. The result demonstrates that our model can consistently outperform many baselines in all evaluation metrics with a large margin.

The rest of the paper is organized as follows. Next, we discuss the preliminaries and framework overview in Section 2. Section 3 describes the method to construct the co-query POI graph from map search query data, as well as the graphlet mining algorithm from the co-query POI graph. Section 4 introduces the details of our proposed neural relationship prediction model. Then we present the experimental results in Section 5. Finally, we discuss the related work in Section 6, and conclude the paper in Section 7 by summarizing our main contributions.

2 Preliminaries and Overview

In this section, we first introduce the preliminaries of our problem, then give a formal description of the competitive relationship prediction problem for POIs. Finally, we present a framework overview to show the work process of the NGP framework. Table 1 lists the important notations used throughout this paper.

TABLE 1
Table of important notations

p_i	id of a POI	$\vec{p}_i$	feature vector of p_i
r_j	id of a region	$\vec{r}_j$	feature vector of r_j
g_a^{se}	a SE-lableled graphlet, a is an index with $1 \leq a \leq m$		
$g_{a,l}^{cse}$	a CSE graphlet genreated from g_a^{se}		
$g_{a,l,i}^{ins}$	an instanced CSE graphlet generated from $g_{a,l}^{cse}$		
FL_a^{se}	a frequent SE-labeled graphlet list for one g_a^{se}		
HL_a^{se}	a hot SE-labeled graphlet list for one g_a^{se}		
$\vec{c}_a$	a graphlet count vector for one g_a^{se}		
$\vec{H}_a$	a hot graphlet feature vector for one g_a^{se}		

2.1 Preliminaries

We use p to denote a POI, and $\vec{p} = (loc, cat, r)$ to denote the feature vector of the POI p, where loc is a 2-dimensional location point on the map, cat denotes the category of the POI (e.g., food, hotel, and bar), and r denotes a region to which the POI p belongs. We divide the entire 2-dimensional space into a set of small regions, and each POI is then classified into the corresponding region given its location. In this work, we divide the space into regions according to the town boundary of a city. The map query data used in this paper, which records users' search behavior, is from Baidu Maps[1]. We formulate map search query data as a set of tuples $D = \{(p, u, ts)\}$, each of which indicates that user u has an interaction (an action on the online maps such as search, click, or view) with POI p at timestamp ts. For a region r, its region feature is denoted as $\vec{r}$ which is constcucted based on the POIs and map search query on these POIs within the region r. According to this feature construction method, region feature $\vec{r}$ has three components: 1) the total number of POIs in the region r; 2) the numbers of POIs in the region r by categories; and 3) the query frequencies of POIs in the region r by categories. Let us denote a set of POIs as $P = \{p_1, p_2, ..., p_n\}$, and use the capital letter with arrow $\vec{P}$ to denote the feature matrix of all the POIs $\vec{P} = [\vec{p}_1^T, \vec{p}_2^T, ..., \vec{p}_n^T]^T$. Similarly, we have region feature matrix $\vec{R} = [\vec{r}_1^T, \vec{r}_2^T, ..., \vec{r}_m^T]^T$. Note that there are several notations about the graphlet in Table 1. To be coherent with the introduction of the graphlet mining algorithm, we postpone the discussion of graphlet related notations in Section 3.2.

The objective of our problem is to associate each POI pair (p_i, p_j) with a label $y \in \{0, 1\}$ where $y = 1$ means p_i and p_j have a competitive relationship. Formally, the POI competitive relationship prediction problem is defined as:

Problem 1. Given a set P of POIs, its corresponding POI feature matrix $\vec{P}$ and region feature matrix $\vec{R}$, and the map search query data D, the objective of our problem is to learn a predictive function $f : (P \times P | \vec{P}, \vec{R}, D) \rightarrow \{0, 1\}$ to predict the competitive relationships between POIs.

2.2 Framework overview

In this section, we present a framework overview of our neural graphlet-based prediction framework (NGP) for competitive prediction on the map search query data. An illustration of our NGP framework is shown in Figure 1. The main process of the NGP framework is to extract graphlets from the co-query POI graph, and then to make a prediction with the features generated from the graphlets. In the framework, first of all, we construct a co-query POI graph from the map query search query data, which is introduced in Section 3.1. Then, as we can see from Figure 1, there are two main components in our NGP framework. The first component is the graphlet mining, which is a specially designed algorithm to mine graphlets from co-query POI graph based on random walk. We introduce the concepts about graphlet in the co-query POI graph in Section 3.2, following by a detailed explanation of the graphlet mining algorithm in Section 3.3. The second component of NGP is the Neural Graphlet-based Relationship Prediction (NGRP) model. In NGRP, we first generate features based on the mined graphlet from the co-query POI graph (see Section 4.1), and then we propose a novel transformer-like structure prediction model with self attention and cross attention mechanism to predict the competitive relationship for each POI pair (see Section 4.2). The NGRP also includes several Graphlet Learning Blocks (GLBs) to process the features generated based on graphlets. Other important factors like region features of POIs, distance between POIs and the count feature based on the frequent graphlets are also investigated by our relationship prediction model.

3 Graphlet Mining on Co-query POI Graph

Graphlets are the basis of our framework to predict competitive POI relationships. We first present how to construct a co-query POI graph from map search query data in Section 3.1. Then, as shown in Figure 2, given a co-query POI graph, NGP first mines many instanced CSE graphlets $g_{a,l,i}^{ins}$ from the graph by random walk (see Figure 2(b)). Then the graphlets $g_{a,l,i}^{ins}$ are inserted into a hash table whose key is a combination of SE-labeled graphlets g_a^{se} and CSE graphlets $g_{a,l}^{cse}$ (see Figure 2(c)). The concepts of $g_{a,l,i}^{ins}, g_{a,l}^{cse}$ *and* g_a^{se} are introduced in Section 3.2. Finally we can construct the feature list FL^{se} and HL^{se} which are used to construct features for relationship prediction in Section 4. The random walk algorithm for the graph mining is introduced in Section 3.3.

3.1 Co-query POI graph construction

The co-query POI graph G is constructed from map search query data and POI data, which encodes the user behavior correlation among POIs reflected in the map search query data. In general, the co-query POI graph $G = (P, E)$ is an undirected graph with P being the set of POIs and $E \subseteq P \times P$ being a set of edges between POIs. Each edge $e_{ij} \in E$ is a tuple $e_{ij} = (p_i, p_j, w_{ij})$ where w_{ij} denotes the edge weight. The weight w_{ij} between p_i and p_j indicates how many users interacting with p_i and p_j in a short time interval recorded in the map search query data. Given a set of map search queries of a user u as $Q_u = \{q_1, q_2, ..., q_L\}$ where each map search query $q_l = (p, u, ts) \in D$. For each query pair $(p_i, u, ts_i) \in Q_u$ and $(p_j, u, ts_j) \in Q_u$, if $|ts_i - ts_j| \leq \Delta T$, there will be a link between p_i and p_j by the user u. The weight w_{ij} of edge e_{ij} is defined as the number of all such links by all users in the map search query D.

1. https://maps.baidu.com/

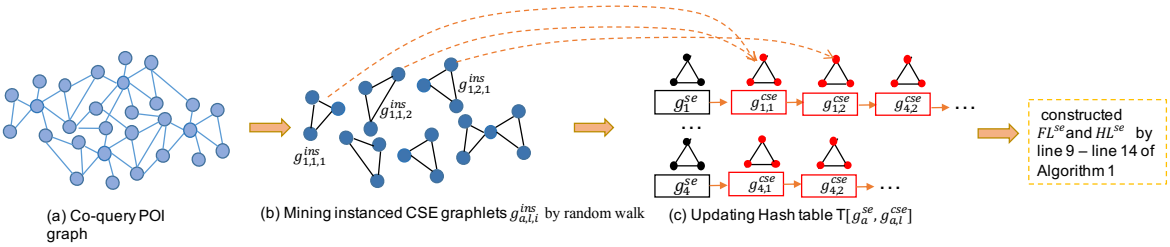


Fig. 2. An overall illustration of graph mining (by Algorithm 1) on co-query POI graph

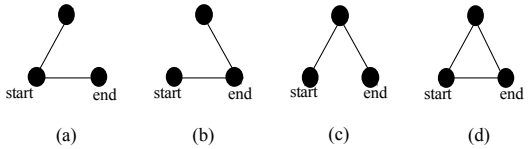


Fig. 3. An illustration of all 3-node SE-labeled graphlets. (a)–(c) are the same if not considering start and end labels.

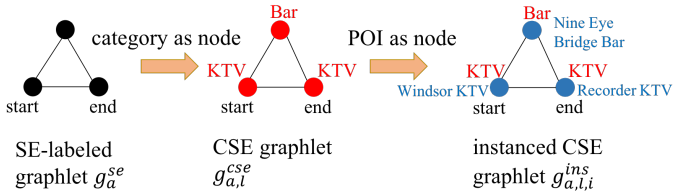


Fig. 4. An example of SE-labeled graphlets, CSE graphlets and instanced CSE graphlets. SE-labeled graphlets have two nodes labeled as start and end; the nodes of CSE graphlets are POI categories; and the ones of instanced CSE graphlets are POIs. Finally, we have $g^{ins}_{a,l,i} \preccurlyeq g^{cse}_{a,l} \preccurlyeq g^{se}_a$.

To reduce the impacts of noisy map search query behaviors, we set a threshold θ_w for edge weights that $e_{ij} \in E$ if and only if $w_{ij} \geq \theta_w$. In our experiment we set weight threshold $\theta_w = 50$ and $\Delta T = 30$ minutes. we give a disucssion about how to set these two parameters, as well as an example of co-query POI graph in Appendix (of the support materials).

3.2 Grpahlet in co-query POI graph

A *graphlet* is a small induced subgraph of a large graph that appears at any frequency (induced subgraphs mean that the subgraph must contain all the edges between nodes appearing in the large graph). We use g to denote a graphlet. Note that a graphlet is different from a motif which is a partial graph without requiring to have all the edges appearing in the subgraph. The first step of our framework is to extract graphlets from the co-query POI graph. To facilitate the relationship prediction, we introduce several concepts about the graphlet in this section, which are SE-labeled graphlet g^{se}_a, Category annotated SE-labeled graphlet (CSE graphlet) $g^{cse}_{a,l}$, and instanced CSE graphlet $g^{ins}_{a,l,i}$.

3.2.1 SE-labeled graphlet

We add a start node label and an end node label on the nodes for each graphlet. With different label positions, the same graphlets may have different forms. We name such labeled graphlets as **SE-labeled graphlets** (SE means start and end), denoted by g^{se}_a, where a is an index, $1 \leq a \leq k$, and k is the number of SE-labeled graphlets. As illustrated in Figure 3, there are 4 SE-labeled graphlets with 3 nodes. Figure 3(a), Figure 3(b) and Figure 3(c) are the same graphlet if without considering the start and end labels. For each SE-labeled graphlet, our objective is to predict the competitive relationship between the start and the end nodes on the graphlet.

3.2.2 CSE graphlet

POI categories play an important role to distinguish the functionalities and services of POIs. When mining graphlets from the co-query POI graph, we further annotate graphlets with the POIs' categories. In this sense, an SE-labeled graphlet might have many variants with different categories on nodes. We call such graphlets as Category annotated SE-labeled graphlets, named as **CSE graphlets** for short. A CSE graphlet generated from SE-labeled graphlet g^{se}_a is then denoted as $g^{cse}_{a,l}$ where l is an index of a set of all CSE graphlets generated from g^{se}_a. We further denote $g^{cse}_{a,l} \preccurlyeq g^{se}_a$ to indicate that $g^{cse}_{a,l}$ is derived from g^{se}_a. We call the relation between SE-labeled graphlet g^{se}_a and CSE graphlet $g^{cse}_{a,l}$ that $g^{cse}_{a,l}$ belongs to g^{se}_a, denoted by $g^{cse}_{a,l} \preccurlyeq g^{se}_a$.

3.2.3 Instanced CSE graphlet

Another type of graphlets is **instanced CSE graphlets**, denoted by $g^{ins}_{a,l,i}$. Each $g^{ins}_{a,l,i}$ is a subgraph of the co-query POI graph that has the same topology structure and POI categories as CSE graphlet $g^{cse}_{a,l}$, where i is another index of the set of all instanced CSE graphlet derived from $g^{cse}_{a,l}$. Similarly, we denote the relation as $g^{ins}_{a,l,i} \preccurlyeq g^{cse}_{a,l}$. We call the relation between CSE graphlet $g^{cse}_{a,l}$ and instanced CSE graphlet $g^{ins}_{a,l,i}$ that $g^{ins}_{a,l,i}$ belongs to $g^{cse}_{a,l}$, denoted by $g^{ins}_{a,l,i} \preccurlyeq g^{cse}_{a,l}$.

To better understand the three types of graphlets and their relations, we illustrate the relation of the SE-labeled graphlet, CSE graphlet and instanced CSE graphlet in Figure 4. To put it in simple words, the nodes of SE-labeled graphlet are just start label, end label, and other graph node; the ones of CSE graphlet are categories of POI and the ones of instanced CSE graphlet are POIs. According to the generation order, we have $g^{ins}_{a,l,i} \preccurlyeq g^{cse}_{a,l} \preccurlyeq g^{se}_a$.

3.3 Graphlet Mining with Random Walks

In this paper, we only consider the graphlet with 3 nodes for relationship prediction. It is possible to incorporate graphlets with more than three nodes into our framework. The major challenge is that frequent graphlet counting is computationally intensive since the number of possible k-graphlets in a graph G increases exponentially [7].

Many previous graphlets papers also focus on mining the graphlets with length 3 (sometimes it is also referred to as graph triangles) like [8]. However, our algorithm can be extended to support graphlet with more nodes such as incorporating the method in [7]. How to efficiently mine graphlets with more than 3 nodes from a large graph is beyond the scope of this paper.

We develop a random walk-based graphlet mining method and present the pseudo codes in Algorithm 1. The first part of Algorithm 1 is graphlet generation (lines 1–8 of Algorithm 1). For a pair of POIs (p_i, p_j), we randomly sample one neighboring POI of p_i or p_j, and then form the corresponding instanced CES graphlet $g^{ins}_{a,l,i}$ and CSE graphlet $g^{cse}_{a,l}$ with 1) replacing the POI node with its category and 2) adding edges of the POI nodes in G to the graphlet. For each possible SE-labeled graphlet with $g^{cse}_{a,l} \preccurlyeq g^{se}_a$, we maintain a hash table $T[g^{se}_a] = \{(g^{cse}_{a,l} : c_{a,l})\}$ to count the number of occurrences of each CSE graphlet where $c_{a,l}$ is the count number. Note that there are 4 SE-labeled graphlets $g^{se}_a (1 \le a \le 4)$ for 3-node graphlet as shown in Figure 3, but there may be many CSE graphlets $g^{cse}_{a,l}$ as shown in Figure 4. We repeat the random sample process to count the appearance of each $g^{cse}_{a,l}$.

The second part of Algorithm 1 is to return two lists of top graphlets as shown in line 9 to line 14. For each SE-labeled graphlet g^{se}_a, we sort all the CES graphlet $g^{cse}_{a,l}$ of $T[g^{se}_a]$ in descending order by its value. Recall that each item in $T[g^{se}_a]$ is a pair of CSE graphlet $g^{cse}_{a,l}$ and the number of all corresponding sampled instanced CSE graphlets $g^{ins}_{a,l,i} \preccurlyeq g^{cse}_{a,l}$. We keep the top-$k$ items for each $T[g^{se}_a]$, and return a *frequent SE-labeled graphlet list* $FL^{se}_a = [g^{cse}_{a,l}], 1 \le l \le k$ where k is the number of frequent CSE graphlets. Besides, we also return the top-k_h items from each $T[g^{se}_a]$, and return a *hot SE-labeled graphlet list* $HL^{se}_a = [g^{cse}_{a,l}], 1 \le l \le k_h$ where k_h is the number of very frequent (i.e., "h" is short for "hot") CSE graphlets with $k_h \ll k$ and HL^{se}_a is a sublist of FL^{se}_a. It is clear that $HL^{se}_a \subset FL^{se}_a$. We return frequent SE-labeled graphlet list set $FL^{se} = \{FL^{se}_a\}, 1 \le a \le m$ and hot SE-labeled graphlet list set $HL^{se} = \{HL^{se}_a\}, 1 \le a \le m$ after the graphlet mining.

Algorithm 1: Graphlet mining with random walks

input : G – a co-query POI graph
output: FL^{se} – frequent SE-labeled graphlet list, HL^{se} – hot SE-labeled graphlet list

1 **repeat**
2 Randomly select a pair of POI (p_i, p_j) from G
3 Randomly select a neighboring POI p_l of p_i or p_j
4 Construct an instanced CSE graphlet $g^{ins}_{a,l,i}$ with nodes p_l, p_i and p_j
5 Form a CSE graphlet $g^{cse}_{a,l}$ by replacing the nodes of $g^{ins}_{a,l,i}$ with POI categories
6 **for** *each SE-labeled graphlet with* $g^{cse}_{a,l} \preccurlyeq g^{se}_a$ **do**
7 $T[g^{se}_a][g^{cse}_{a,l}] \leftarrow T[g^{se}_a][g^{cse}_{a,l}] + 1$ // $T[g^{se}_a]$ is a hash table
8 **until** *sample enough pairs*
9 **for** *each* g^{se}_a *in Figure 3* **do**
10 Sort all key-valve pairs $(g^{cse}_{a,l}, c_{a,l})$ of $T[g^{se}_a]$ in descending order of $c_{a,l}$
11 Construct frequent SE-labeled graphlet list $FL^{se}_a \leftarrow T[g^{se}_a][1:k]$ // $FL^{se}_a = [g^{cse}_{a,l}], 1 \le l \le k$
12 Construct hot SE-labeled graphlet list $HL^{se}_a \leftarrow T[g^{se}_a][1:k_h]$ // $HL^{se}_a = [g^{cse}_{a,l}], 1 \le l \le k_h$ and $k_h \ll k$
13 $FL^{se} \leftarrow FL^{se} + FL^{se}_a$
14 $HL^{se} \leftarrow HL^{se} + FL^{se}_a$
15 Return FL^{se}, HL^{se}

4 Neural Graphlet-based Relationship Prediction Model

In this section, we introduce how to utilize the graphlet as well as the POI and region features for competitive relationship prediction of POI pairs, then introduce the proposed neural relationship prediction model.

4.1 Graphlet Feature Generation

We utilize the graphlet list set FL^{se} and HL^{se} to generate features for our NGP model. The pseudo code for feature generation is shown in Algorithm 2.

4.1.1 Graphlet count feature generation

Our first step is to generate a graphlet count feature matrix $\vec{C}$ based on FL^{se} for a candidate POI pair. For each SE-labeled graphlet g^{se}_a, we first construct a hash table T_a, in which each key corresponds to one CSE graphlet in FL^{se}_a (see line 4 of Algorithm 2). Given a candidate POI pair (p_i, p_j) to predict the competitive relationship, we randomly select a neighboring POI of p_i or p_j in the co-query POI graph, and then construct an instanced CSE graphlet $g^{ins}_{a,l,i}$ (see line 6 to line 7 of Algorithm 2). Then we iterate all CSE graphlets $g^{cse}_{a,l}$ in the frequent SE-labeled graphlet list FL^{se}_a. If $g^{ins}_{a,l,i}$ could be derived from a CSE graphlet $g^{cse}_{a,l}$ in FL^{se}_a (i.e., $g^{ins}_{a,l,i} \preccurlyeq g^{cse}_{a,l}$ and $g^{cse}_{a,l} \in FL^{se}_a$), we increase the value of $T_a[g^{cse}_{a,l}]$ by 1 (see line 8 to line 10 of Algorithm 2). In our experiments, we repeat the random sample iteration for 1,000 times for each pair. After repeating the random sample process many times, we construct a graphlet count vector $\vec{c}_a$ based on T_a, where the value of each dimension of $\vec{c}_a$ equals to the count value of $T_a[g^{cse}_{a,l}]$. Since the number of CSE graphlets $g^{cse}_{a,l}$ for each g^{se}_a in FL^{se} is k, the dimension of $\vec{c}_a$ is also k, i.e. $\vec{c}_a \in \mathbb{R}^k$. Finally, we can generate a graphlet count feature matrix $\vec{C} = [\vec{c}_1, \ldots, \vec{c}_a, \ldots, \vec{c}_m]^T$, where m is the total number of SE-labeled graphlets, and $\vec{C} \in \mathbb{R}^{k \times m}$.

4.1.2 Hot graphlet feature generation.

The second step is to extract the hot graphlet feature from HL^{se} for a candidate POI pair. Given a candidate POI pair (p_i, p_j), for generating $\vec{C}$ we have already constructed many instanced CSE graphlets $g^{ins}_{a,l,i}$ (see line 6 to line 7 of Algorithm 2). Meanwhile, we also iterate all CSE graphlets $g^{cse}_{a,l}$ in the hot SE-labeled graphlet list HL^{se}_a. If the instanced CSE graphlet $g^{ins}_{a,l,i}$ could be derived from $g^{cse}_{a,l}$ in HL^{se}_a (i.e., $g^{ins}_{a,l,i} \preccurlyeq g^{cse}_{a,l}$ and $g^{cse}_{a,l} \in HL^{se}_a$), we will generate a hot graphlet feature vector for $g^{cse}_{a,l}$ based on $g^{ins}_{a,l,i}$ (see line 12 to line line 16 of Algorithm 2). Note that for each $g^{cse}_{a,l} \in HL^{se}_a$, we only use one instanced CSE graphlet $g^{ins}_{a,l,i} \preccurlyeq g^{cse}_{a,l}$ to generate the feature vector, therefore we remove the matched $g^{cse}_{a,l}$ from HL^{se}_a after generating the feature (see line 14 of Algorithm 2).

Algorithm 2: Graphlet feature generation

input : G – a co-query POI graph,
(p_i, p_j) – a POI pair,
FL^{se} – frequent SE-labeled graphlet list set,
HL^{se} – hot SE-labeled graphlet list set
output: $\vec{C}$ – graphlet count feature matrix for POI pair (p_i, p_j),
$\vec{H}$ – hot graphlet feature matrix for POI pair (p_i, p_j)

1 **for** *each* g_a^{se} *shown in Figure 3* **do**
2 Get FL_a^{se} from FL^{se}
3 Get HL_a^{se} from HL^{se}
4 $T_a \leftarrow ()$ // T_a is a hash table with key as CSE graphlet and value as count integer
5 **repeat**
6 Randomly select a neighbor node p_l of p_i or p_j from G
7 Construct an instanced CSE graphlet $g_{a,l,i}^{ins}$.
8 **for** $g_{a,l}^{cse} \in FL_a^{se}$ **do**
9 **if** $g_{a,l,i}^{ins} \preccurlyeq g_{a,l}^{cse}$ **then**
10 $T_a[g_{a,l}^{cse}] += 1$
11
12 **for** $g_{a,l}^{cse} \in HL_a^{se}$ **do**
13 **if** $g_{a,l,i}^{ins} \preccurlyeq g_{a,l}^{cse}$ **then**
14 $HL_a^{se} \leftarrow HL_a^{se} - g_{a,l}^{cse}$
15 Construct feature vector $\vec{h}_{a,l}$ from $g_{a,l,i}^{ins}$ that $\vec{h}_{a,l} = \vec{h}_{a,l}^{p} \oplus \vec{h}_{a,l}^{w} \oplus \vec{h}_{a,l}^{\xi}$, where $\vec{h}_{a,l}^{p} = \vec{p}_i \oplus \vec{p}_j \oplus \vec{p}_l$, $\vec{h}_{a,l}^{w} =< w_{ij}, w_{i,l}, w_{j,l} >$ and $\vec{h}_{a,l}^{\xi} =< \xi_{ij}, \xi_{i,l}, \xi_{j,l} >$
16 $\vec{H}_a \leftarrow \vec{H}_a \cup \{\vec{h}_{a,l}\}$ // $\vec{h}_{a,l} \in \mathbb{R}^{d_h}, 1 \le l \le k_h, \vec{H}_a \in \mathbb{R}^{k_h \times d_h}$
17
18 **until** *sample n times* // $n = 1000$ in our experiment
19 Construct count vector $\vec{c}_a \in \mathbb{R}^k$ from T_a // each dimension of $\vec{c}_a$ corresponds to a $g_{a,l}^{cse}$ in T_a.
20 $\vec{C} = \{\vec{c}_a\}$ // $1 \le a \le m$
21 $\vec{H} = \{\vec{H}_a\}$ // $1 \le a \le m$
22 **return** $\vec{C}, \vec{H}$

More specifically, for each CSE graphlet $g_{a,l}^{cse}$, we represent its feature as $\vec{h}_{a,l}$ which consists of three parts (see line 15 of Algorithm 2), including: 1) a POI composition feature by composing the feature vectors of POIs in the instanced CSE graphlet $g_{a,l,i}^{ins}$ which can be expressed as $\vec{h}_{a,l}^{p} = \vec{p}_i \oplus \vec{p}_j \oplus \vec{p}_l$ where $\vec{p}_l$ is the third POI in $g_{a,l,i}^{ins}$, 2) a vector of edge weights between POIs in the co-query POI graph $\vec{h}_{a,l}^{w} = (w_{ij}, w_{il}, w_{jl})$, and 3) a vector of distances between the POIs $\vec{h}_{a,l}^{\xi} = (\xi_{ij}, \xi_{il}, \xi_{jl})$ where $\xi_{ij} = \text{dist}(p_i, p_j)$. The feature vector $\vec{h}_{a,l}$ is obtained by concatenating the three feature vectors, i.e., $\vec{h}_{a,l} = \vec{h}_{a,l}^{p} \oplus \vec{h}_{a,l}^{w} \oplus \vec{h}_{a,l}^{\xi}$. We suppose the total length of feature vector $\vec{h}_{a,l}$ being d_h (i.e., $\vec{h}_{a,l} \in \mathbb{R}^{d_h}$), and the number of CSE graphlets $g_{a,l}^{cse}$ for each g_a^{se} in HL^{se} is k_h. Then, for each SE-labeled graphlet g_a^{se}, we can generate a feature matrix $\vec{H}_a = [\vec{h}_{a,1}, \ldots, \vec{h}_{a,l}, \ldots, \vec{h}_{a,k_h}]^T$. The size of feature matrix $\vec{H}_a$ is $k_h \times d_h$ (i.e., $\vec{H}_a \in \mathbb{R}^{k_h \times d_h}$). Since we generate a feature matrix $\vec{H}_a$ for each SE-labeled graphlet $g_a^{se} (1 \le a \le m$, see line 1 of Algorithm 2), the final hot graphlet feature set is $\vec{H} = \{\vec{H}_a \mid 1 \le a \le m\}$.

Time complexity. There are three loops in Algorithm 2. The first loop (from line 1 to line 19) runs m times, and the seond loop (from line 5 to line 18) runs n times. The two inner loops (line 8-10 and line 12-16) run k and k_h times respectively. It seems that the time time complexity of Algorithm 2 is $O(m(n(k+k_h)))$. However, in our setting, m=4 (there are 4 g_a^{se}) and n=1000, both of them are constant, therefore the time complexity of Algorithm 2 is actually $O(k + k_h)$, whose time complexity is still acceptable in applications.

4.2 Relationship prediction model

Our devised NGRP model has a transformer-like structure [9] with self attention and cross attention to fully utilize the graphlet-based features. An illustration of NGRP model is shown in Figure 5. Before explaining the NGRP model, we first summarize the featues for relationship prediction. Given a candidate POI pair (p_i, p_j) for relationship prediction, we have the following features: POI feature $\{\vec{p}_i, \vec{p}_j\}$ and their region feature $\{\vec{r}_i, \vec{r}_j\}$; the graphlet count feature $\vec{C} = \{\vec{c}_a\}$ and hot graphlet feature $\vec{H} = \{\vec{H}_a\}, 1 \le a \le m$.

As we can see from Figure 5, an important component of the NGRP model is the Graphlet Learning Block (denoted by GLB for short) for each SE-labeled graphlet g_a^{se}. As introduced in Section 4.1.2, for an SE-labeled graphlet g_a^{se}, we have its hot graphlet feature as $\vec{H}_a \in \mathbb{R}^{k_h \times d_h}$. In the GLB, at first, we apply a dense layer transform on $\vec{H}'_a = \vec{H}_a W^i, W^i \in \mathbb{R}^{d_h \times d_o}$. After that, the single head self attention on the feature matrix as:

$$\vec{\alpha}_{s,a} = softmax(\vec{H}'_a \vec{W}_s^Q (\vec{H}'_a \vec{W}_s^K)^T)(\vec{H}'_a \vec{W}_s^V) \quad (1)$$

where $\vec{W}_s^Q \in \mathbb{R}^{d_o \times d_s}, \vec{W}_s^K \in \mathbb{R}^{d_o \times d_s}, \vec{W}_s^V \in \mathbb{R}^{d_o \times d_s}$ *and* $\vec{\alpha}_{s,a} \in \mathbb{R}^{k_h \times d_s}$. The specific meanings of K, Q and V actually follow the self-attention mechanism of transformer, where Q represents the "query" vector, K represents the "key" vector, and V represents the "value" vectors. The self-attention score will be computed by the dot products of the query vector $(\vec{H}'_a \vec{W}_s)$ and key vector $(\vec{H}'_a \vec{W}_s^K)$ with a softmax function outside of the dot product. Then the final output of the model is the multiplication of each value vector $(\vec{H}'_a \vec{W}_s^V)$ by the attention score. With the self-attention score, GLB can pay more attention to the discriminative features generated from graphlets. Thus, the self attention mechanism can help GLB more efficiently utilize the features generated from the graphlets. In our model we conduct multi-head self attention of $\vec{H}'_a$:

$$\vec{\Lambda}_{s,a} = (\vec{\alpha}_{s,a}^1 \oplus ... \oplus \vec{\alpha}_{s,a}^h) W_s^m \quad (2)$$

where $\oplus$ is catenation operation, and $W_s^m \in \mathbb{R}^{h \cdot d_s \times d_o}$.

Then we compute the feature matrix of $\vec{H}'_a$ after self-attention with a residual connection:

$$\vec{H}''_a = \vec{H}'_a \circ \vec{\Lambda}_{s,a} + \vec{H}'_a \quad (3)$$

where $\circ$ is Hadamard product. We also apply a layer normalization on $\vec{H}''_a$.

For each POI, we can obtain a feature vector by the concatenation of POI feature of $\vec{p}$ and its region vector $\vec{r}$, which is expressed as: $\vec{z} = \vec{p} \oplus \vec{r}$ and $\vec{z} \in \mathbb{R}^{d_z}$. For

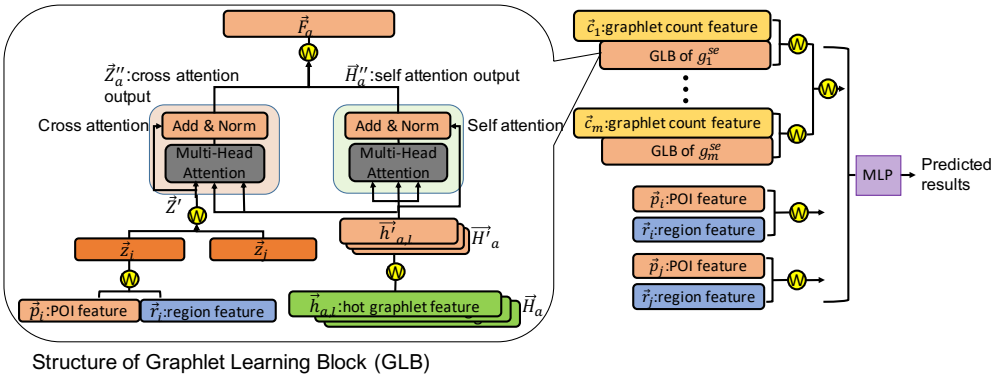


Fig. 5. An illustration of neural graphlet-based relationship prediction (NGRP) model.

a pair of POI, we denote the POI pair feature matrix of these two POIs as $\vec{Z} \in \mathbb{R}^{2\times d_z}$. We first apply a dense layer transformation $\vec{Z}' = \vec{Z}W^j, W^j \in \mathbb{R}^{d_z\times d_o}$. Then we compute a cross attention between $\vec{Z}'$ and $\vec{H}'_a$ whose single head cross attention is:

$$\vec{\alpha}_{c,a} = softmax(\vec{Z}'\vec{W}_c^Q(\vec{H}'_a\vec{W}_c^K)^T)(\vec{H}'_a\vec{W}_c^V) \quad (4)$$

$\vec{W}_c^Q \in \mathbb{R}^{d_o\times d_s}, \vec{W}_c^K \in \mathbb{R}^{d_o\times d_s}, \vec{W}_c^V \in \mathbb{R}^{d_o\times d_s}$ *and* $\vec{\alpha}_{c,a} \in \mathbb{R}^{2\times d_s}$. The cross-attention score is computed by the dot products of the query vector $(\vec{Z}'\vec{W}_c^Q)$ with key vector $(\vec{H}'_a\vec{W}_c^K)$ with a softmax function outside of the dot product. Then the final output of the model is the multiplication of each value vector $(\vec{H}'_a\vec{W}_c^V)$ by the attention score. We can also conduct the multi-head cross attention of $\vec{Z}'$ and $\vec{H}'_a$:

$$\vec{\Lambda}_{c,a} = (\vec{\alpha}^1_{c,a} \oplus ... \oplus \vec{\alpha}^h_{c,a})W_c^m \quad (5)$$

where $\oplus$ is catenation operation, and $W_c^m \in \mathbb{R}^{h\cdot d_s\times d_o}$.

The intuition of cross attention is that the query vector $(\vec{Z}'\vec{W}_c^Q)$ can affect the feature important of graphlet features generated from the hot graphlet feature $\vec{H}$. Therefore, with the help of cross attention, the GLB can adaptively learn the graphlet features for the relationship prediction when they are in different contexts depending on the POIs and regions. For example, for the same graphlet features, the competitive relationship for different POI pairs may still be different if they are located in downtown or in suburban district. The region/POI features are encoded again outside of the GLB to utilize such features for final prediction.

Then we can get the POI pair feature matrix $\vec{Z}$ under the SE-labeled graphlet g_a^{se} after cross attention with a residual connection:

$$\vec{Z}''_a = \vec{Z}' \circ \vec{\Lambda}_{c,a} + \vec{Z}' \quad (6)$$

The layer normalization is also employed on $\vec{Z}''_a$.

At the end of GLB, for each g_a^{se} we concatenate $\vec{H}''_a$ and $\vec{Z}''_a$ together to output a final feature matrix $\vec{F}_a$ that:

$$\vec{F}_a = (\vec{H}''_a \oplus \vec{Z}''_a)W_a^f \quad (7)$$

where $W_a^f \in \mathbb{R}^{d_o\times d_f}$ and $\vec{F}_a \in \mathbb{R}^{(2+d_a)\times d_f}$.

As shown in Figure 5, outside of the GLB, there are two features for each SE-labeled graphlet g_a^{se}: graphlet attention feature matrix $\vec{F}_a$ generated by each GLB, and graphlet count feature vector $\vec{c}_a$. We flat the matrix $\vec{F}_a$ and then concatenate it with $\vec{c}_a$ to form the SE-labeled graphlet feature vector $\vec{l}_a = (flatten(\vec{F}_a) \oplus \vec{c}_a)W^l, W^l \in \mathbb{R}^{((2+d_a)d_f+k)\times d_l}$. The overall output of the graphlet feature is $\vec{g} = (\vec{l}_1 \oplus ... \oplus \vec{l}_m)W^g, W^g \in \mathbb{R}^{(m\cdot d_l)\times d_g}$.

Finally, we concatenate $\vec{g}$ and $\vec{Z}'$, and input all the features into a Multilayer Perceptron (MLP) to make the binary prediction: $\hat{y}_{ij} = MLP(\vec{g} \oplus flatten(\vec{Z}'))$. We use $ReLu(\cdot)$ as the activation function, minimize the cross entropy to optimize whole model with Adam optimizer [10]. Note that the selection of the final classifier is independent of the NGP model. MLP is adopted because it has a universal approximation property to approximate the optimal Bayes discriminant function. Using better-designed network architecture to replace MLP may improve the performance, but it is beyond the scope of this paper.

5 Experiments

In this section, we conduct an extensive experimental evaluation on real-world data sets to demonstrate the effectiveness of our algorithm. At first, we present our dataset and the process to construct the ground truth. Then we compare our framework with several state-of-the-art baselines to show its effectiveness. Finally, we conduct parameter evaluation and ablation study of different model components.

5.1 Experimental setting

We evaluate our model on two city-level datasets BEIJING and CHENGDU. To construct the POI graphs, we use the map search query data and POI data in the corresponding cities from August 1, 2018 to August 31, 2018. The data is obtained from Baidu Maps, one of the largest commercial online map service platform in China. There are 307K POIs and 320 regions in BEIJING, and 235K POIs and 177 regions in CHENGDU. We keep all the map searches and regions in the cities. The regions are divided by the town boundaries. After manually removing the POI which cannot provide service to users (i.e. exclude the categories like road, doors, and hill etc.), there are 86 categories used in the datasets such as supermart, theater, cinema and cafe. How to set the parameters of θ_w and ΔT is discussed in Section 3.1. The statistics of our datasets are summarized in Table 2.

5.1.1 *Ground truth*

Here we introduce a method to construct the ground-truth POI competitive relationships, by combining user check-in

TABLE 2
Statistics of datasets

Dataset	POIs	Regions	Categories
BEIJING	307K	320	86
CHENGDU	235K	177	86

data and knowledge graph data. The user check-in data is also provided by the online map service platform from July 1, 2018 to July 31, 2018. The knowledge graph data is a public knowledge base zhishi.me [11][2]. Our construction process has three steps. First, we only keep POIs with brands. Second, we search all brand names in zhishi.me to find their related brands which have a "relatedPage" relation. Third, under each brand pair, for all pairs of POIs between the two brands (like a store of KFC and a store of McDonald's), we further filter the POI pairs by the following conditions: i) the distance between the two POIs is within 10 km; ii) they have the same category; and iii) the fraction of common check-in users during July 2018 is larger than 5%. Finally, there are 18,731 pairs of competitive POIs in BEIJING and 7,514 pairs in CHENGDU. We also randomly select 200 pairs from BEIJING and CHENGDU, respectively, and manually check the accuracy of the sampled pairs. We find that on both datasets the accuracy is larger than 95%, which indicates the correctness of our method to construct the ground truth.

All information used to construct the ground truth has been excluded from the input features of POIs. First, we use different time windows for user check-in data and map search query data to avoid information leakage. Second, we do not use the name and other text descriptions of POIs as features, since zhishi.me might employ such text information to construct entity relations. Thus, both the user check-in data and text data are not used to extract features in our framework. From this perspective, our method can be considered as a relationship completion method based on the seed POI competitive relationships mined from user check-in data and text data. Note that map search query data has a much larger amount and better coverage than user check-in data.

5.1.2 Negative sampling

We also construct the same number of POI pairs as negative, i.e., non-competitive, samples. We generate the negative samples with similar rules of ground truth construction: 1) the distance between POIs is within 10 km; 2) they have the same category; 3) the distribution of the numbers of POIs in different categories is consistent with the positive pairs. The number of the negative sample is equal to the number of positive pairs on both datasets.

5.1.3 Data splitting

We randomly split the datasets into 80% for training, 10% for validation, and 10% for testing. In all tests, if without clarification, the best parameters are optimized on validation data, and the experimental performance on testing data with the chosen parameters is reported.

2. http://zhishi.me/

5.2 Baselines

We adopt the Precision (Prec), Recall (Rec), F1-measure (F1), Accuracy (Acc), and Area under the curve (AUC) as evaluation metrics. We compare our model with the following baselines, which are heuristic methods (**DIST** (by distance), **check-in** (by check-in), **EW** (graph edge weight), **PA** (preferential attachment [12]), **CN** (common neighbors [13]), **RA** (resource allocation [14]), **JC** (Jaccard [15])), classification methods(**MLP** (Multilayer Perceptron) and **XGBoost** [16]), **PRA** and **PRA+fea** [17], **Node2vec** and **node2vec+fea** [18], and **GNN models** (GNN-SEAL [19], Geom-GCN [20] and GIN [21]). The detailed introduction of the baselines can be found in Appendix (of the support matrials).

5.3 Overall performance evaluation

We compare all baselines as our NGP framework on both BEIJING and CHENGDU datasets. The results are presented in Table 3. We can see that NGP outperforms all other models with a significant margin in all metrics on both BEIJING and CHENGDU. Note that we do not show the F1, Precision and Recall for all heuristic methods since such rule-based methods usually lead to a very high precision but very low recall (or very high recall but very low precision), whose metrics are not reasonable to compare with other methods.

At first, NGP outperforms all heuristic rule-based models (i.e. DIST, EW, PA, CN, RA and JC) with a significant margin in all metrics on BEIJING and CHENGDU datasets as shown in Table 3. The accuracy of DIST is only 0.6442, which means that simply determining the competitive relationship with distance is not reasonable. All heuristic methods, including EW, PA, CN, RA and JC, do not perform well for relationship prediction since the co-query graph is noisy. It is hard to infer a relationship directly with some heuristic rules for link prediction. Taking EW as an example, we consider a pair of POIs to be competitive if their co-query weight is higher than a threshold (and they have the same category). However, as shown in Table 3, the accuracy of EW is only 0.5023, which is much worse than NGP (which is 0.8243). This shows that POI's relationship on a co-query graph is complex, and it is difficult to learn POI's relationship well through general rule-based and unsupervised methods. It also verifies our claim in the Introduction section that if the edge weight of the co-query POI graph is large, there are two opposite facts for competitive relationship. Therefore we need an advanced model to explore such useful but noisy information.

NGP also outperforms classification methods including MLP and XGBoost with feature engineering. Note that XGBoost can also be viewed as an "EW with adaptive thresholds" method, which adaptively sets thresholds with rules defined by trees. From Table 3 we can see that the AUC of NGP is higher than MLP by 14.0% (0.7918→0.9030) and XGBoost by 9.2% (0.8273→0.9030) on BEIJING. This is because, though we extract some features from the data for MLP and XGBoost to predict competitive relationships, the manual feature engineering efforts still cannot fully utilize the hidden patterns in the co-query POI graph.

NGP can also achieve better performance than the graph-based methods (PRA+fea, node2vec+fea, GNN-

TABLE 3
Overall performance evaluation

Method	BEIJING					CHENGDU				
	Acc	AUC	F1	Prec	Rec	Acc	AUC	F1	Prec	Rec
DIST	0.6453	0.8118	/	/	/	0.6067	0.6731	/	/	/
EW	0.5023	0.7047	/	/	/	0.5068	0.3064	/	/	/
PA	0.5001	0.4948	/	/	/	0.5010	0.5041	/	/	/
CN	0.5046	0.5433	/	/	/	0.533	0.6086	/	/	/
RA	0.5225	0.5552	/	/	/	0.5025	0.4391	/	/	/
JC	0.5000	0.4973	/	/	/	0.5020	0.5172	/	/	/
Check-in	0.7253	0.7883	/	/	/	0.6786	0.7593	/	/	/
MLP	0.7191	0.7918	0.7302	0.6946	0.7697	0.6537	0.7333	0.6467	0.6761	0.6197
Xgboost	0.7554	0.8273	0.7483	0.7608	0.7362	0.6804	0.7581	0.6581	0.7264	0.6015
PRA	0.7250	0.7927	0.6906	0.777	0.6216	0.6591	0.7215	0.6444	0.6904	0.6041
PRA+fea	0.7622	0.8361	0.7623	0.7795	0.7458	0.7190	0.8082	0.7121	0.7478	0.6796
node2vec	0.6892	0.7438	0.6685	0.7063	0.6345	0.6831	0.7485	0.6860	0.6951	0.677
node2vec+fea	0.7860	0.8642	0.7906	0.8080	0.7739	0.7549	0.8284	0.7526	0.7777	0.7291
GNN-SEAL	0.8037	0.8798	0.8040	0.8143	0.7940	0.7605	0.8384	0.7574	0.7784	0.7375
GIN	0.795	0.8729	0.7972	0.8066	0.7881	0.7597	0.844	0.7657	0.747	0.7853
Geom-GCN	0.8059	0.8795	0.8031	0.8211	0.7859	0.7636	0.8493	0.7603	0.7712	0.7497
NGP	**0.8243**	**0.9030**	**0.8296**	**0.8232**	**0.8361**	**0.7776**	**0.8556**	**0.7828**	**0.7818**	**0.7838**

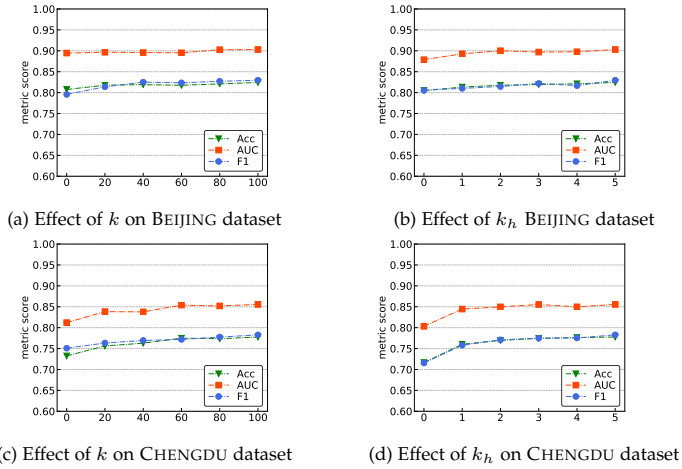
SEAL, GIN and Geom-GCN). On BEIJING, the AUC of NGP is larger than PRA+fea by 8.0% (0.8361→0.9030), node2vec+fea by 4.5% (0.8642→0.9030), GNN-SEAL by 2.6% (0.8798→0.9030), GIN by 3.4% (0.8729→0.9030) and Geom-GCN by 2.7% (0.8729→0.9030). Since the co-query POI graph is noisy, node2vec+fea may learn a distorted representation of the graph; while our NGP framework utilizes a graphlet mining method to filter the noisy information. For PRA+fea, the graphlet structures contain richer information than paths of PRA. For the GNN-based models (GNN-SEAL, GIN and Geom-GCN), the co-query graph is much sparser than general graphs, and they cannot fully captures the patterns in the co-query graph for link prediction. Therefore, NGP achieves the best prediction performance for competitive relationship prediction among all approaches.

Since the competitive relationship prediction is a fundamental problem for business, the significance of the performance improvement of our model can be explained from two application perspectives. On the one hand, the predicted relationship can help a business owner to obtain a more comprehensive understanding of the competitive environment. A little accuracy improvement can give the business owner a competitive edge against his/her competitors. On the other hand, the competitive relationship can be used as features for prediction models, such as the advertising prediction model for online location-based advertisement, one percent accuracy improvement on relationship prediction can potentially bring considerable revenue for advertisement service providers.

5.4 Parameter evaluation

Here we evaluate the effect of parameters: number of frequent CSE graphlets k and number of hot CSE graphlets k_h on BEIJING and CHENGDU datasets.

To test the effect of the number of frequent CSE graphlets k, we vary k from 0 to 100 in steps of 20. It should be noted that the sequence has been sorted according to the number of graphlet occurrence times, so the sequence we used is always the one with the most occurrences. It can be seen from Figure 6(a)(c) that when the sequence is not used (the number of frequent CSE graphlets k is zero), the evaluation score decreases obviously. When k becomes 20 from 0, all the metrics arise significantly. But when $k \geq 20$, all the metrics become relatively stable. This indicates that the model is not sensitive with varying the value of k.

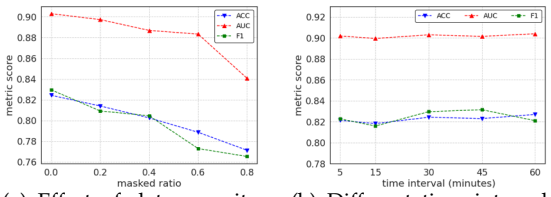


(a) Effect of k on BEIJING dataset
(b) Effect of k_h BEIJING dataset
(c) Effect of k on CHENGDU dataset
(d) Effect of k_h on CHENGDU dataset

Fig. 6. Parameter evaluation on BEIJING and CHENGDU datasets.

We also vary k_h from 0 to 5 to reveal the impact of the number of hot CSE graphlets. Similarly, the graphlet we use is extracted from more to less graphlets after sorting this type. Figure 6(b)(d) show that all metrics arise when k_h is increased from 0 to 2. When no graphlet (the number of hot CSE graphlets k_h is zero) is used, the score of evaluation is much lower than that of others. When $k_h \geq 3$, all the metrics become relatively stable. The experiment demonstrates that the graphlet feature is useful, but NGP model is not sensitive to parameters related to the graphlets when k and k_h are large enough.

The sparsity of the datasets also has an impact on the performance of NGP. As we can see from Table 3, the prediction preformance of NGP on BEIJING dataset is better than the one on CHENGDU dataset. One possible reason for this phenomenon is that the data density of Beijing is larger than Chengdu. We also conducted an experimental evaluation as



(a) Effect of data sparsity on BEIJING (b) Different time interval on BEIJING

Fig. 7. Varing dataset settings on BEIJING.

shown Figure 7(a) by randomly masking (removing) some percent of edges of the whole co-query POI graph to illustrate the change of the performance of NGP. As we can see fromFigure 7(a), with varying the percent of masked edges, the prediction performance of NGP gradually declines.

We also conduct an experimental evaluation on Beijing dataset with varying different time intervals. As we can see from Figure 7(b), with varying the time interval, the prediction performance of NGP is relative stable. A possible reason for this phenomenon is that different users in different scenarios have different time intervals for searching POIs, and NGP can always learn the competitive relationships of POIs if there is enough information indicated in a certain time interval by users.

5.5 Ablation study of model components

In this section, we conduct an ablation study to demonstrate the effectiveness of each component. The results are shown in Table 4. The graphlet method, attention, graphlet count feature, and hot graphlet feature are the key components of the NGP model. To test the effect of these components, we remove one of them separately and use the remaining parts to make predictions on both BEJING and CHENGDU datasets under the accuracy, AUC and F1-measure.

The results show that, if anyone of them is removed, the evaluation measures will be reduced to different extents on the datasets. This proves that all the components have a great impact on the model results, and all the components are indispensable. First, we conclude that "graphlet" is useful compared with "path" on the co-query POI graph. This can be verified by comparing the NGP_path (NGP model with paths instead of graphlets) and NGP in Table 4. As we can see from Table 4, using graphlets instead of paths, NGP can achieve about 3.8% improvement in accuracy metric on BEJING (0.7939→0.8243) and 6.2% improvement on CHENGDU dataset (0.7323→0.7776). A good description "signature" extracted from co-query POI graph between two POIs is important to improve the prediction performance. Compared with NGP_path, the NGP with graphlets can better exploit the topological structures of the co-query POI graph.

Second, we demonstrate the effectiveness of the graphlet count feature and hot graphlet feature. In Table 4, NGP_no_HGF refers to NGP without using the hot graphlet feature, and NGP_no_GCF refers to NGP without using graphlet count feature. As we can see, both the graphlet count feature and the hot graphlet feature can improve the performance of NGP. Third, we also evaluate the effectiveness of the attention mechanism (self attention and cross attention). Third, NGP_no_CrossAtt means we use simple equal weights to replace the cross attention weights and NGP_no_SelfAtt means we use equal weights to replace the self attention weights in NGP. Table 4 shows that the attention mechanism can bring improvement in metrics of Accuracy, AUC and F1-score. Note that in Table 4, sometimes NGP cannot achieve the best results for precision and recall. However, NGP can always obtain the best performance under the F1-measure metric which a more fair metric considering both the precision and the recall.

Third, we verify the effect of POI and region features inside and outside of GLB in Table 4. The NGP_no_glb_poi and NGP_no_glb_region show the performance of NGP after removing POI and region features of GLB respectively, and NGP_no_cat_feat shows the performance of removing POI and region features outside of GLB. As we can see from Table 4, removing the POI/region features either of them can reduce the performance of NGP. To sum up, the results in Table 4 prove that all the components have a positive impact on model performance, and all of them are indispensable.

6 RELATED WORK

The research topic of this paper is closely related with competitive relationships mining and link prediction. Spatial keyword search is also related with our research which returns POIs from user queries [22]. How to return the POI from the spatial keyword search is beyond the scope of paer, and we refer readers to a comprehensive survey [23].

6.1 Competitive relationships mining

There are already some studies about competitive relationship mining on text data, which is pioneered by [3]. However, most of them focus on mining competitive relationships among companies or products, without touching the problem for POIs. Authors in [24] study how to extract comparative opinions from the reviewers' comment data. Whereas, authors in [25] propose a Topic Factor Graph Model for detecting competitors which not only considers the text information, but also incorporates the social network information. However, as discussed in [4], such comparative evidence is typically scarce, or even non-existent in many domains. There are a few of studies to extract comparative relationship from text data and heterogeneous data. For example, some competition mining tasks first identify comparative sentences, and then extracts the entities being compared within these sentences [5], [25], [26], [27]. There is also a recent work to extract POI competitive relationships from both review (text) data and map query data [28]. However, many unpopular POIs lack of such text data where this method cannot be applied in the task for POIs with only map search query data. Meanwhile, our method make perdiction solely based on map search query data.Zhang et al. [29] investigate a dedicated company embedding model by learning network embeddings over fine-grained multiple talent flow networks formed by different person roles and job positions. Their method cannot be used for POI competitive relationship prediction since the POI graph does not have such multi-flow information (of different person roles

TABLE 4
Effect of each component of the NGP model

Method	BEIJING					CHENGDU				
	Acc	AUC	F1	Prec	Rec	Acc	AUC	F1	Prec	Rec
NGP_path	0.7939	0.8729	0.7957	0.7792	0.8129	0.7323	0.8101	0.7280	0.7577	0.7005
NGP_no_HGF	0.8049	0.8787	0.8056	0.8211	0.7907	0.7324	0.8120	0.7509	0.7163	**0.7890**
NGP_no_GCF	0.8075	0.8946	0.7956	**0.8704**	0.7327	0.7165	0.8033	0.7156	0.7282	0.7034
NGP_no_CrossAtt	0.8161	0.8939	0.8094	0.8466	0.7753	0.7606	0.8319	0.765	0.7751	0.7551
NGP_no_SelfAtt	0.8099	0.8928	0.8008	0.7588	**0.8478**	0.7497	0.8299	0.7493	0.7318	0.7678
NGP_no_glb_poi	0.8062	0.8908	0.8151	0.7957	0.8355	0.7491	0.8296	0.7569	0.7439	0.7703
NGP_no_glb_reg	0.8118	0.8982	0.8067	0.8493	0.7682	0.7523	0.8462	0.7552	0.7632	0.7473
NGP_no_cat_feat	0.8191	0.8994	0.8218	0.8275	0.8163	0.7618	0.8356	0.7705	0.7531	0.7887
NGP	**0.8243**	**0.9030**	**0.8296**	0.8232	0.8361	**0.7776**	**0.8556**	**0.7828**	**0.7818**	0.7838

and job positions). Lu et al. [30] propose an algorithm to mine a well-defined competitive relation between a pair of two spatial object groups (instead of a pair of two spatial objects), which is a different problem from ours. Another direction of related work is the method proposed in [31] which tries to rank businesses by user check-in data. This method, however, is significantly different from ours. This is not only because their method is unsupervised, but also because the objective is to rank businesses instead of extracting the competitive relationships among the businesses.

6.2 Link prediction

There has been a tremendous amount of work on link prediction in the past decade in various fields. Generally, the methods can be divided into topological feature-based and latent feature-based methods [32]. A class of topological feature-based methods is to extract rules and inductive logics to infer new links [33], [34]. An extension of such topological feature-base methods is the path ranking algorithm and its variants [35], [36]. More recently, researchers adopt the deep learning method, like Restricted Boltzmann Machine [37], Weisfeiler-Lehman Neural Machine [32] and graph neural network [19], [38], for link prediction. We refer readers to a comprehensive survey for a summary of link prediction methods [39], [40]. In our paper, we focus on how to make the competitive relationship prediction between POIs, where we should consider the POI features and the auxiliary edges among the POIs extracted from map search query data. Most of these existing link prediction methods only use the graph structure to predict the missing links, whereas our problem is to predict unobserved relationships between POIs with map search query data. To sum up, as far as we have known, existing methods for competitive relationship prediction, such as texting mining method and link prediction, cannot be used to solve this problem directly.

7 CONCLUSION

In this paper, we studied competitive relationship prediction for POIs. By exploiting the large-scale online map search query data from a POI graph perspective, we presented a neural graphlet-based prediction (NGP) framework for POI competitive relationship prediction. After building a co-query POI graph from map search query data, the NGP framework adopts a graphlet mining method to exploit the patterns on the co-query POI graph, and then a neural graphlet-based relationship prediction (NGRP) model was designed to make prediction. According to the experimental results, our framework could outperform all the baselines in all metrics, including accuracy, AUC, precision, recall, and F1-measure.

ACKNOWLEDGMENTS

This research is supported in part by grants from the National Natural Science Foundation of China (91746301, 71531001) and K.C.Wong Education Foundation.

REFERENCES

[1] S. Yang, M. Wang, W. Wang, Y. Sun, J. Gao, W. Zhang, and J. Zhang, "Predicting commercial activeness over urban big data," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 3, p. 119, 2017.
[2] H. Cao, Z. Chen, F. Xu, Y. Li, and V. Kostakos, "Revisitation in urban space vs. online: a comparison across pois, websites, and smartphone apps," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 4, p. 156, 2018.
[3] N. Jindal and B. Liu, "Mining comparative sentences and relations," in *AAAI*. AAAI Press, 2006, pp. 1331–1336.
[4] T. Lappas, G. Valkanas, and D. Gunopulos, "Efficient and domain-invariant competitor mining," in *KDD*. ACM, 2012, pp. 408–416.
[5] W. Kessler and J. Kuhn, "Detection of product comparisons-how far does an out-of-the-box semantic role labeling system take you?" in *EMNLP*, 2013, pp. 1892–1897.
[6] C. Zhuang, N. J. Yuan, R. Song, X. Xie, and Q. Ma, "Understanding people lifestyles: Construction of urban movement knowledge graph from gps trajectory." in *IJCAI*, 2017, pp. 3616–3623.
[7] N. K. Ahmed, J. Neville, R. A. Rossi, and N. Duffield, "Efficient graphlet counting for large networks," in *ICDM*. IEEE, 2015, pp. 1–10.
[8] C. E. Tsourakakis, U. Kang, G. L. Miller, and C. Faloutsos, "Doulion: counting triangles in massive graphs with a coin," in *KDD*. ACM, 2009, pp. 837–846.
[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017, pp. 5998–6008.
[10] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[11] X. Niu, X. Sun, H. Wang, S. Rong, G. Qi, and Y. Yu, "Zhishi. me-weaving chinese linking open data," in *ISWC*. Springer, 2011, pp. 205–220.
[12] M. E. Newman, "Clustering and preferential attachment in growing networks," *Physical review E*, vol. 64, no. 2, p. 025102, 2001.
[13] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *Journal of the American society for information science and technology*, vol. 58, no. 7, pp. 1019–1031, 2007.
[14] T. Zhou, L. Lü, and Y.-C. Zhang, "Predicting missing links via local information," *The European Physical Journal B*, vol. 71, no. 4, pp. 623–630, 2009.
[15] G. Salton and M. J. McGill, "Introduction to modern information retrieval," 1986.
[16] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *KDD*. ACM, 2016, pp. 785–794.

[17] M. Gardner and T. Mitchell, "Efficient and expressive knowledge base completion using subgraph feature extraction," in *EMNLP*, 2015, pp. 1488–1498.
[18] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *KDD*. ACM, 2016, pp. 855–864.
[19] M. Zhang and Y. Chen, "Link prediction based on graph neural networks," in *NIPS*. ACM, 2018.
[20] H. Pei, B. Wei, K. C.-C. Chang, Y. Lei, and B. Yang, "Geom-gcn: Geometric graph convolutional networks," in *ICLR*, 2020.
[21] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in *ICLR*, 2019.
[22] X. Cao, L. Chen, G. Cong, C. S. Jensen, Q. Qu, A. Skovsgaard, D. Wu, and M. L. Yiu, "Spatial keyword querying," in *International Conference on Conceptual Modeling*. Springer, 2012, pp. 16–29.
[23] L. Chen, S. Shang, C. Yang, and J. Li, "Spatial keyword search: a survey," *GeoInformatica*, vol. 24, no. 1, pp. 85–106, 2020.
[24] K. Xu, S. S. Liao, J. Li, and Y. Song, "Mining comparative opinions from customer reviews for competitive intelligence," *Decision support systems*, vol. 50, no. 4, pp. 743–754, 2011.
[25] Y. Yang, J. Tang, J. Keomany, Y. Zhao, J. Li, Y. Ding, T. Li, and L. Wang, "Mining competitive relationships by learning across heterogeneous networks," in *CIKM*. ACM, 2012, pp. 1432–1441.
[26] R. Li, S. Bao, J. Wang, Y. Yu, and Y. Cao, "Cominer: An effective algorithm for mining competitors from the web," in *ICDM*. IEEE, 2006, pp. 948–952.
[27] X. Ding, B. Liu, and L. Zhang, "Entity discovery and assignment for opinion mining applications," in *KDD*. ACM, 2009, pp. 1125–1134.
[28] S. Li, J. Zhou, T. Xu, H. Liu, X. Lu, and H. Xiong, "Competitive analysis for points of interest," in *KDD*, 2020, pp. 1265–1274.
[29] L. Zhang, T. Xu, H. Zhu, C. Qin, Q. Meng, H. Xiong, and E. Chen, "Large-scale talent flow embedding for company competitive analysis," in *WebConf*, 2020, pp. 2354–2364.
[30] J. Lu, L. Wang, Y. Fang, and M. Li, "Mining competitive pairs hidden in co-location patterns from dynamic spatial databases," in *PAKDD*. Springer, 2017, pp. 467–480.
[31] T.-N. Doan, F. C. T. Chua, and E.-P. Lim, "Mining business competitiveness from user visitation data," in *SBP*. Springer, 2015, pp. 283–289.
[32] M. Zhang and Y. Chen, "Weisfeiler-lehman neural machine for link prediction," in *KDD*. ACM, 2017, pp. 575–583.
[33] S. Muggleton, "Inverse entailment and progol," *New generation computing*, vol. 13, no. 3-4, pp. 245–286, 1995.
[34] L. A. Galárraga, C. Teflioudi, K. Hose, and F. Suchanek, "Amie: association rule mining under incomplete evidence in ontological knowledge bases," in *WWW*. ACM, 2013, pp. 413–422.
[35] N. Lao and W. W. Cohen, "Relational retrieval using a combination of path-constrained random walks," *Machine learning*, vol. 81, no. 1, pp. 53–67, 2010.
[36] N. Lao, T. Mitchell, and W. W. Cohen, "Random walk inference and learning in a large scale knowledge base," in *EMNLP*. Association for Computational Linguistics, 2011, pp. 529–539.
[37] X. Li, N. Du, H. Li, K. Li, J. Gao, and A. Zhang, "A deep learning approach to link prediction in dynamic networks," in *SDM*. SIAM, 2014, pp. 289–297.
[38] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *arXiv preprint arXiv:1901.00596*, 2019.
[39] M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich, "A review of relational machine learning for knowledge graphs," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 11–33, 2016.
[40] V. Martínez, F. Berzal, and J.-C. Cubero, "A survey of link prediction in complex networks," *ACM Computing Surveys*, vol. 49, no. 4, p. 69, 2017.

Jingbo Zhou is a staff research scientist at Business Intelligent Lab of Baidu Research, working on machine learning problems for both scientific research and business applications, with a focus on spatial temporal data mining, user behavior study and knowledge graphs. He obtained his Ph.D. degree from National University of Singapore in 2014, and B.E. degree from Shandong University in 2009. He has published several papers in top venues, such as SIGMOD, KDD, VLDB, ICDE, TKDE and AAAI.

Tao Huang is a researcher on the search recommendation team of 58.com. His research focuses on deep learning and recommender system. He received his master's degree from University of Electronic Science and Technology of China in 2020, and BE degree from Harbin Institute of Technology in 2016. He had been a research intern at Business Intelligent Lab of Baidu Research.

Shuangli Li is working toward his Master's degree in Computer Application Technology at the University of Science and Technology of China. He received his BS degree in computer science and technology from University of Science and Technology of China in 2019. He had been a research intern at Business Intelligence Lab, Baidu Research. His research focuses on graph representation learning, recommender system and knowledge graph.

Renjun Hu received the BE degree from Beihang University in 2014. He is working toward the PhD degree in the School of Computer Science and Engineering, Beihang University. He was a visiting student at Rutgers, The State University of New Jersey and a research intern at Business Intelligent Lab, Baidu Research. His research focuses on graph algorithms and applied machine learning, with a special interest on developing contextual representation learning methods for mobile analytics.

Yanchi Liu received the PhD in Information Technology from Rutgers, the State University of New Jersey and the PhD in Management Science from the University of Science and Technology Beijing. His research interests include data mining, artificial intelligence, business intelligence, urban computing, and recommender systems. He has published prolifically in top conferences/journals such as KDD, IJCAI, WWW, TCYB. He also served as program committee members among top conferences in the data mining and artificial intelligence communities.

Dr. Yanjie Fu is an assistant professor in the Department of Computer Science at the University of Central Florida. He received his Ph.D. degree from Rutgers, the State University of New Jersey in 2016, the B.E. degree from University of Science and Technology of China in 2008, and the M.E. degree from Chinese Academy of Sciences in 2011. His research interests include data mining and big data analytics.

Hui Xiong is a Professor at the Rutgers, the State University of New Jersey. Xiong's research interests include data mining, mobile computing, and their applications in business. Xiong received his PhD in Computer Science from University of Minnesota, USA. He has served regularly on the organization and program committees of numerous conferences, including as a Program Co-Chair of the Industrial and Government Track for the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), a Program Co-Chair for the IEEE 2013 International Conference on Data Mining (ICDM), a General Co-Chair for the 2015 IEEE International Conference on Data Mining (ICDM), and a Program Co-Chair of the Research Track for the 2018 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. He received the 2021 AAAI Best Paper Award and the 2011 IEEE ICDM Best Research Paper award. For his outstanding contributions to data mining and mobile computing, he was elected an AAAS Fellow and an IEEE Fellow in 2020.