



# RELINe: point-of-interest recommendations using multiple network embeddings

Giannis Christoforidis<sup>1</sup> · Pavlos Kefalas<sup>1</sup> · Apostolos N. Papadopoulos<sup>1</sup> · Yannis Manolopoulos<sup>2</sup>

Received: 25 July 2019 / Revised: 22 December 2020 / Accepted: 23 December 2020 /

Published online: 13 January 2021

© The Author(s), under exclusive licence to Springer-Verlag London Ltd. part of Springer Nature 2021

## Abstract

The rapid growth of users' involvement in Location-Based Social Networks has led to the expeditious growth of the data on a global scale. The need of accessing and retrieving relevant information close to users' preferences is an open problem which continuously raises new challenges for recommendation systems. The exploitation of points-of-interest (POIs) recommendation by existing models is inadequate due to the sparsity and the cold start problems. To overcome these problems many models were proposed in the literature; however, most of them ignore important factors, such as: geographical proximity, social influence, or temporal and preference dynamics, which tackle their accuracy while personalize their recommendations. In this work, we investigate these problems and present a unified model that jointly learns user's and POI dynamics. Our proposal is termed RELINe (**RE**commendations with **mu**LtIple Network **E**MBEDdings). More specifically, RELINe captures: (i) the *social*, (ii) the *geographical*, (iii) the *temporal influence*, and (iv) the *users' preference dynamics*, by embedding eight relational graphs into one shared latent space. We have evaluated our approach against state-of-the-art methods with three large real-world datasets in terms of accuracy. Additionally, we have examined the effectiveness of our approach against the cold-start problem. Performance evaluation results demonstrate that significant performance improvement is achieved in comparison to existing state-of-the-art methods.

**Keywords** POI recommendations · Location-based social networks · Temporal evolution · Geographical dynamics · Social influence

---

✉ Giannis Christoforidis  
icchrist@csd.auth.gr

Pavlos Kefalas  
kefalasp@csd.auth.gr

Apostolos N. Papadopoulos  
papadopo@csd.auth.gr

Yannis Manolopoulos  
yannis.manolopoulos@ouc.ac.cy

<sup>1</sup> Aristotle University of Thessaloniki, Thessaloniki, Greece

<sup>2</sup> Open University of Cyprus, Latsia, Cyprus

# 1 Introduction

Recently, *Online Social Networks* (OSN) incorporated geographical information into their content which triggered new functionalities and introduced the concept of *Location-based Social Networks* (LBSNs). In such networks, such as Facebook Places<sup>1</sup>, Foursquare<sup>2</sup>, Yelp<sup>3</sup>, users may share their interests along with spatial dimension to obtain recommendations for possibly interesting places based on their recent history. Learning users' history is a crucial task for these models, to provide meaningful suggestions for *Points of Interest* (POIs). Unfortunately, factors like sparsity, heterogeneity and multidimensionality pose significant challenges that increase the problem complexity with a large impact on effectiveness and efficiency.

There has been extensive research on the topic, which primarily focused on users' relations with geographical information over user-location bipartite networks. However, such approaches failed to eliminate the sparsity problem since they miss preference dynamics and auxiliary information related to users' interactions. Similarly, other works enrich user-location information with social network ties but still miss all aforementioned factors. Moreover, research considering the temporal evolution of users' preferences still fail to deal with sparsity because there is additional contextual information related to users which changes along with their preferences over time.

In this paper, we present RELINE (**RE**commendations with **mu**LtIple **N**etwork **E**mbddings), a technique that considers social, geographical and temporal influence, along with users' preference dynamics, in a unified model, embedding eight relational graphs into a single latent space. More specifically, the information taken into account contains:

*Spatial Behavior.* Recent research points out that there is a spatial pattern on users' check-in behavior during daily activities [4]. In particular, users who check-in a location within a region have high probability to attend locations that are proximate. For example, a user located close to work or home has higher probability to visit a proximate location, rather than one in a long distance [10]. We claim what users tend to perform a sequence of activities during the same time period within a region which is related to a task. For example, users who want to do shopping usually check-in a mall, a supermarket, a grocery, or a bakery located close to home, as shown in Fig. 1a. Also, during weekends people perform multiple check-ins in clubs and restaurants that are close to each other. Thus, spatial proximity should be considered as a repeated geographical pattern. Also, the relation of those locations can be considered as a route of locations that are related to one activity.

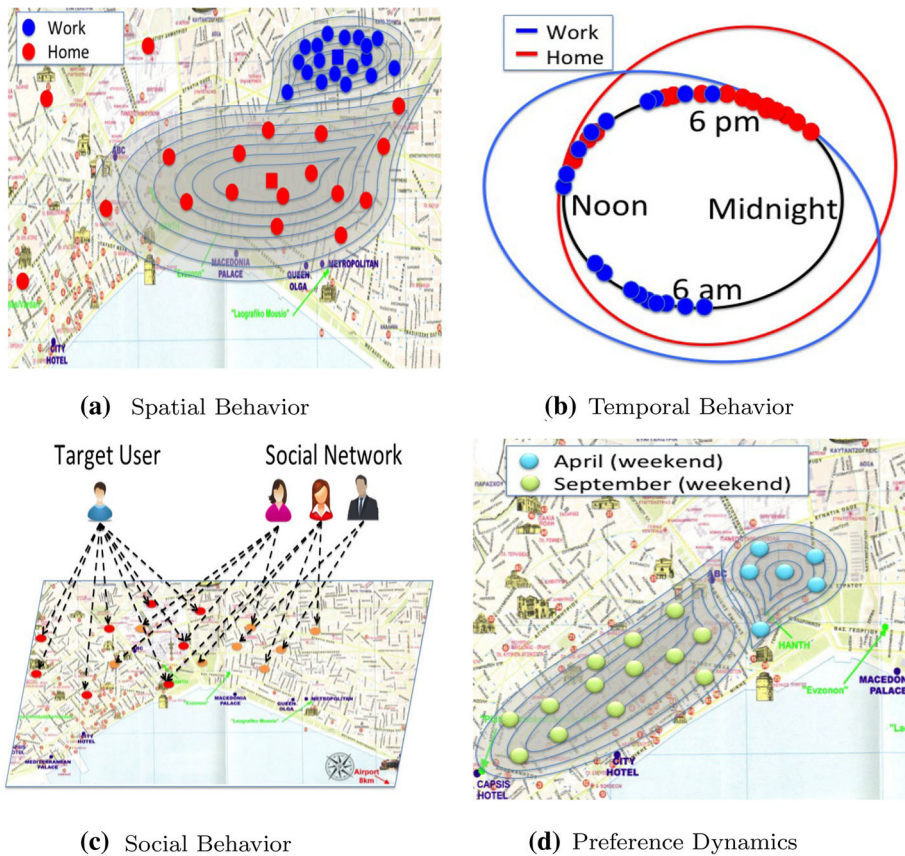
*Temporal Behavior* Usually, users maintain a fixed daily program in their activities and the check-ins they perform. Thus, on weekdays a user performs check-ins at locations close to work 6 – 5, whereas from 5 p.m. until midnight she check-ins at locations close to home, as shown in Fig. 1b. This pattern is reversed in weekends because users tend to check-in bars and restaurants. Several works aim to model this behavior by focusing on temporal drifting but they ignore explicit and implicit contextual information [17,22] related to users activities.

*Preference Dynamics* Users tend to change behavior during time which makes this problem even harder. For example, a user may attend clubs every weekend of September and cinemas or restaurants during April weekends, as shown in Fig. 1d. In both cases, the

<sup>1</sup> [www.facebook.com/places](http://www.facebook.com/places).

<sup>2</sup> [www.foursquare.com](http://www.foursquare.com).

<sup>3</sup> [www.yelp.com](http://www.yelp.com).



**Fig. 1** **a** Spatial distribution of one user's check-ins close to home or work, **b** temporal distribution of check-ins where both lines indicate the proportional probability of being in one of these two states, **c** the social behavior indicate the influence of the social network to the target user, and **d** preference dynamics evolution between two months

same user alternates his check-in behavior, which should be taken into account. This preference evolution according to [10,26] may be due to: (1) *New POIs exploration*: contrary to ordinary check-in behavior, users tend to visit new locations, (2) *User experience*: users will choose a location according to locations in which he had a pleasant experience in the past, (3) *Popularity*: some locations tend to be more popular during a time interval rather than the rest of the year, (4) *Social influence*: friends' opinion bias users' decisions (as shown in Fig. 1c). Thus, users tend to examine their friends' evaluations over locations before attending and follow their lead.

**Motivation and Contribution** Based on the research conducted so far in the area, there are several limitations, such as:

- many methods consider POIs as conventional nodes and do not capture the spatial dimension and the proximity among them,
- although some works consider the geographical influence, they ignore preference dynamics tackling the over all accuracy of the solution,

- methods that capture temporal dynamics do not treat simultaneously the spatial dimension,
- finally, models that consider both spatial and temporal behavior ignore the preference evolution.

Thus, there is a need to consider all aforementioned factors in a unified model which allows to further understand users' behavior and personalize the recommendations. The contributions of our work are summarized as follows:

- We present a probabilistic weighting strategy over the 8 information graphs to overcome sparsity.
- We propose two novel algorithms to extract the routes and the stay points out of the past history check-ins.
- A novel graph-based approach is introduced that jointly learns users' and POI embeddings from these weighted graphs into the same latent space and provides personalized POI recommendations.
- Our approach extends significantly the LINE model [21] and learn the embeddings of large unipartite and bipartite graphs simultaneously into a low dimensional space. It is noted that LINE can only capture first and second order proximity and it is used only for embeddings. In contrast, RELINE is more general and also it is designed to provide POI recommendations.
- Our work surpasses other heterogeneous embedding techniques and multi-view learning techniques by extending the learning process and simultaneously learning from multiple networks.
- RELINE is experimentally evaluated measuring the accuracy of POI recommendations for: (i) all users, (ii) cold-start users, and (iii) cold-start locations.

*Extensions beyond the Conference Version* This work is an extended version of our work presented at the 5th IEEE International Conference on Data Science and Advanced Analytics (DSAA 2018) [5]. This journal version contains several enhancements with respect to the conference paper. The most significant changes are summarized below:

- We introduce two novel networks: (i) *Stay Points*, representing locations the user stayed the most, and (ii) *Routes*, the path followed when visiting POIs.
- Two novel algorithms are used to build the aforementioned information networks which we weigh according to their importance.
- We jointly capture users' and POIs sequential dynamics.
- The performance evaluation section has been extended significantly by studying two important topics in the domain related to the cold-start problem for both users and locations.
- RELINE is compared against additional state-of-the-art methods in terms of accuracy.

*Roadmap.* The rest of the paper is organized as follows. Section 2 summarizes the related work, whereas Sect. 3 describes the preliminaries and the problem definition. In Sect. 4, we illustrate the model structural parts in details, while in Sect. 5, we present the findings of our experimentation. Finally, Sect. 6 concludes this paper and discusses briefly future work in the area.

## 2 Related work

Here, we discuss research related to POI recommendation. In particular, we analyze how the prior work exploits  $n$ -dimensional networks to overcome sparsity and cold start problems.

These networks include social ties, geographical proximity, temporal distance, and preference dynamics of users' check-in past history.

**POI Recommendations** The lack of direct relation between trajectory points and users' preferences derived from their check-in records tackled the research of that direction. Recently, with the raise of LBSNs, users are able to check-in locations, resulting in anonymous access of their data for research purposes. In this context, there has been a lot of interest in recommendation models that use such information. Most of the related work uses Collaborative Filtering (CF) [30], Content-based Filtering (CB) [6,18] or hybrid [10,25,28] approaches to learn users' preferences over attended POIs and makes predictions for unvisited locations. The former approaches offer recommendations based on the assumption that users who visit the same POIs are most likely to visit the same locations in the future. For instance, Yuan et al. exploit the similarity among users through the check-in history and use collaborative filtering [30]. On the other hand, content-based approaches use additional information related to users or POIs to tackle sparsity problem. Similarly, Gao et al. [6] exploit the content information of LBSNs by investigating the types of content information related to POI attributes along with check-in records. To overcome problems each method faces separately, such as: (i) treating POIs as nodes and ignoring the geographical proximity, and (ii) missing additional dimensions such as social influence or temporal evolution, hybrid approaches were introduced. Below we discuss in detail models that use additional information networks.

*Social Influence.* Since check-in records cannot always overcome the sparsity and cold-start problems, many approaches use social ties following the assumption that users tend to follow their friends' lead [10]. For example, Li et al. [13] distinguish three types of friendships, that are: (i) linked, (ii) co-located, and (iii) proximate friends and exploit their check-in records through a unified framework. First, this model learns the common POIs that the target user and all three types of friends check-in the past. Then, it uses matrix factorization to minimize two loss functions over the learned POIs to personalize recommendations. Similarly, Zhang et al. [31] introduced another unified model named LORE (Location Recommendations) that combines sequential patterns with social and geographical influence. This model exploits the sequential influence of POIs over users' records through a dynamic graph with Additive Markov Chain (AMC). Finally, it combines all aforementioned influences into one model with product fusion rule equation.

*Geographical Influence* To further enhance the knowledge about users and eliminate sparsity, many methods use geographical information. Unfortunately, some of them treat POIs as conventional items [16] and miss the geographical proximity between locations. On the other hand, recent studies [23,24,28,29] consider the geographical proximity and treat them as 'spatial items'. In particular, Wang et al. [24] focused on the importance of sequential influence of spatial items a user purchases, and proposed a *Sequential Personalized item REcommender* system (SPORE) that fuses the sequential influence of spatial items and the preferences of a target user into the same latent space through a probabilistic topic-region unified model. Extending previous work related to spatial item recommendation, another probabilistic model [23] was introduced that jointly correlates the geographical influence, item attributes, and users' reviews into one unified framework named LSARS. Both models support the claim that users are willing to interact with proximate items, thus they use geographical influence jointly with additional factors.

Yin et al. [29] claimed that users keep the same preferences either being in their hometown or when they visit a new region. The authors used spatial attributes of POIs to alleviate sparsity and cold-start problems for out-of-town users. The geographical influence, such as region attributes, is used as auxiliary information to auto-correlate users with new locations. That

model, named SH-CDL, jointly learns these attributes along with additive representations of users' check-in preferences. This way they search for proximate items close to users' past preferences in other regions. In the same content, Yang et al [28] developed a *Preference And Context Embedding* (PACE) model that jointly learns the embeddings of users and locations. This hybrid model bridges Semi-Supervised Learning (SSL) with Collaborative Filtering (CF) in a unified way.

*Temporal Influence* Recent studies indicate that there is a periodicity on users' behavior which should also be encountered along with all aforementioned factors [10,15,32]. Yuan et al. [30] examined the spatiotemporal behavior during different hours of day claiming that users tend to visit specific locations at different time periods. They proposed a unified model that combines CF and Bayes rule to provide POI recommendations considering both proximity and periodicity. Similarly, Kefalas et al. [10] proposed another hybrid model that combines the CF and CB in a unified way exploring: (i) the proximate users' preferences, (ii) the textual influence alternation within time periods, and (iii) the preference dynamics evolution. Results indicate an evolution of users' check-in behavior, since they are highly influenced by all factors combined. The results pointed an incremental robustness of the precision against models considering each factor separately. In the same direction, Liu et al. [15] developed another hybrid spatiotemporal-aware model that learns the jointly representations of users, spatiotemporal patterns and POIs.

On the other hand, following the embedding models like word2vec [19], Zhao et al. [32] presented Geo-Temporal sequential embedding rank (Geo-Teaser) that combines both: (i) the temporal POI embeddings, and (ii) the spatial hierarchical pairwise ranking. In particular, the first half model learns POI representations based on a temporal POI embedding model which uses users' daily check-in history as a sequence. The second half model ranks POIs based on geographical information in a hierarchical pair-wise preference. Similarly, Xie et al. [25] presented a unified graph-based model named GE that jointly learns POI embeddings by capturing the sequential effect, the spatial influence, the periodicity and the semantics into the same latent space.

In contrast to previous work, we propose a methodology that considers all the aforementioned factors into one unified model. In particular, we apply a probabilistic strategy to weigh the importance of edges over eight information graphs. Then, we introduce a graph-based approach that jointly learns users' and POIs embeddings from these weighted graphs into the same latent space and provides personalized POI recommendations. Furthermore, we examine the performance of our approach against *cold-start users* and *cold-start POIs* problems in terms of accuracy. The proposed technique shows significant performance improvement in comparison to existing state-of-the-art methods. To the best of our knowledge, this is the first attempt in literature to face both problems simultaneously.

*Learning embeddings in shared low-dimensional representation* Learning embeddings from multiple sources is a problem which attracted a lot of interest during the last decades [27]. In particular, the main problem faced is the heterogeneity of data related to the main entity which need to be represented in the same latent space. Chang et al. [2] proposed a method which determines similarities between nodes in a heterogeneous network. In particular, the two data types are represented using linear transformation matrices and similarities can be identified by calculating their inner product. The correlations between nodes can be parameterized by the projection matrices or by a bilinear function which allows their close proximity to be represented through closeness while disconnected nodes remain isolated. However, their work supports only undirected graphs and similarities are represented only through node correlations. Our work, supports first and second order proximity between nodes, resulting in a greater number of hops in the learning process.



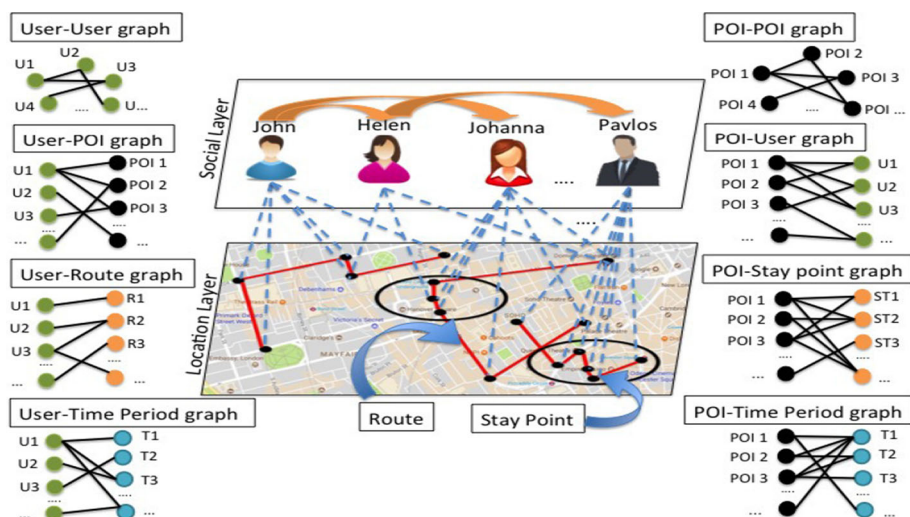


Fig. 2 An example of participating graphs in an LBSN

Following a similar approach, Chen et al. [3] transform input graphs into embeddings in order to calculate the closeness between nodes via their Euclidean distance with aiming at clustering similar nodes together. A loss function is used to minimize the distance between linked nodes along with negative sampling for scaling. In contrast to this research, our work uses the bidirectionality of the nodes with different weights to negative sample the graph instead of sampling from both sides of an edge.

Many clustering models were developed [7,8,11] in order to learn multiview features in a shared latent space. Those works attempt to capture the contextual correlations so that they can identify the similarities between nodes. Our approach differs from multiview features as we update all embeddings dynamically according to vertices in the graphs. The simultaneous training applied allows for better a grasp of the users' preferences along with the locations' properties and time information. Additionally, the discarded features MLVs algorithms use in order to reduce the dimensionality, may proven to be counter-productive especially on the cold start cases, both for users and locations.

### 3 Preliminaries and problem definition

In this section, we present the problem definition in detail and also discuss some fundamental concepts related to our research. Figure 2 depicts all participating networks, whereas Table 1 presents the most frequently used symbols in the sequel. Some fundamental definitions follow.

**Definition 1** (*POI*) is a unique location in which users checked-in. A POI is represented as a tuple:  $\langle l_{id}, \text{longitude}, \text{latitude} \rangle$ .

**Definition 2** (*Check-in*) is a self-report positioning of a user  $u$ , in a location  $l$ , at time  $t$ , represented as a tuple:  $c_i = \langle u, l, t \rangle$ . One check-in can be performed only by one user, but the same user may have multiple check-ins recorded in the profile  $c_{u_i} = \{ \langle l_i, t_i \rangle, \dots, \langle l_j, t_j \rangle \}$ .

**Table 1** Frequently used symbols

Symbol	Description
$U, L, T, R, C$	Set of users $U = \{u_1, \dots, u_n\}$ , Set of locations $L = \{l_1, \dots, l_m\}$ , Set of Time intervals $T = \{t_1, \dots, t_k\}$ , Set of Routes $R = \{r_1, \dots, r_p\}$ , Set of Check-ins $C = \{c_1, \dots, c_o\}$
$c_{u_i}$	User's check-in
$l_{u_i, r}$	Location belong to user's route
$r_{num}$	Number of POIs in routes
$\Delta T$	Time interval
$S, N$	Number of samples, number of negative samples
$e_{i, j}$	Edge between two nodes
$E$	Set of edges $e_{i, j}$ over each graph
$w_{i, j}$	Weighted edge $e_{i, j}$
$W$	Set of weights $w_{i, j}$ over each graph
$S$	An edge $e_{i, j}$ sampled from $W$

**Definition 3** (*Time Period*) is defined as distinct time intervals of the entire dataset, divided in equal size bins of days, weeks or months. Each period contains the check-ins of all users performed during that time interval  $\Delta T$ .

**Definition 4** (*User-User graph*) is a unipartite graph that describes the social network of the users. This graph is denoted as  $\mathcal{G}_{uv} = (U \cup V, \mathcal{W}_{uv})$ , where  $U$  and  $V$  are sets of users, and  $\mathcal{W}_{uv}$  is the set of the weighted edges among them. This network is an undirected graph of friendship relations among users. Thus, the bidirectional connection between the vertices is described as  $\langle u, v \rangle = \langle v, u \rangle$  and the weight is defined as the fraction of 1 over the number of users ( $v$ ), who are friends with user  $u$ :

$$\mathcal{W}_{uv} = \frac{1}{\sum_{i=1}^n |v_i|}$$

**Definition 5** (*User-POI graph*) is a directed bipartite graph which indicates relation between users and locations for the entire check-in history. In particular, it defines the importance of one location for a target user against all the other locations. This graph is denoted as  $\mathcal{G}_{ul} = (U \cup L, \mathcal{W}_{ul})$ , where  $U$  is the set of users,  $L$  is the set of locations, and  $\mathcal{W}_{ul}$  is the set of the weighted edges among them. The weights are computed as the fraction of the number of times a user  $u$  visited location  $l$  against all over all her check-ins:

$$\mathcal{W}_{ul} = \frac{\sum_{\forall c_{u_i} \in l_i} c_{u_i, l}}{\sum_{\forall c_{u_i} \in L} c_{u_i, L}}$$

**Definition 6** (*User-Time period graph*) is a directed bipartite graph representing the interaction of a user at a time period. This graph is denoted as  $\mathcal{G}_{ut} = (U \cup T, \mathcal{W}_{ut})$ , where  $U$  is the set of users,  $T$  is the set of time periods, and  $\mathcal{W}_{ut}$  is the set of the weighted edges among them. The weights are computed as the fraction of the number of each user check-ins during



**Algorithm 1:** ROUTEEXTRACTION

---

**Input:**  $C$ : Set of users' check-ins history,  
 $\Delta T$ : time interval  
**Output:**  $R$ : a list of routes of all users'

```

1 Sort  $C$  based on timestamp
2 Create  $T$  periods with time interval  $\Delta T$ 
3 Create list of Routes  $R$ 
4 for ( $i=0, i==T, i++$ ) do                                // For every route  $t_i \in T$ 
5      $n \leftarrow 0$ ;
6     while ( $T_{start} \leq c_{u_i}[n] \leq T_{end}$ ) do          // While  $c[n] \in T_i$ 
7          $T[i] \leftarrow c_{u_i}[n]$ ;
8          $n++$ ;
9      $q \leftarrow 0$ ;
10    for ( $j=0, j==sizeof(T[i], j++)$ ) do                  // For every route  $l \in T_i$ 
11        if ( $u_i[j+1] == u_i[j]$ ) then                      // By the same user  $u$ 
12             $R[q].add(c_{u_i}[n])$ 
13        else
14             $q++$ ;
15 return  $R$ 
    
```

---

a time period against all the check-ins s/he has made:

$$\mathcal{W}_{ut} = \frac{\sum_{\forall c_{u_i} \in T_i} c_{u_i,t}}{\sum_{\forall c_{u_i} \in T} c_{u_i,T}}$$

**Definition 7** (*Route*) is the sequence of POIs each user attended during a time period  $\Delta T$ . For example  $\{t_1 : l_1 \rightarrow l_2 \rightarrow l_3\}$  indicates that user  $u$  moved from location  $l_1$  to location  $l_2$  and then ended his route at location  $l_3$  during time period  $t_1$ . To extract the routes of each user, we first sort the check-ins according to time. Then, we split the dataset into time periods and for each user we construct the route sequence during that time interval.

**Definition 8** (*User-Route graph*) is a directed bipartite graph which emphasizes the importance of a route for each user. First, we extract the routes from given POIs (as described in Definition 1) with Algorithm 1. Then, we correlate each user from  $U$  to one route from the set  $R$ . This relation is represented as  $\mathcal{G}_{ur} = (U \cup R, \mathcal{W}_{ur})$ . The weight of the edge between two nodes is computed as the number of times one user followed one particular route to the total number of routes made by the same user during all time periods:

$$\mathcal{W}_{ur} = \frac{\sum_{\forall l_{u_i,t} \in T_i} c_{u_i,t}}{\sum_{\forall l_{u_i,t} \in T} R_{u_i,T}}$$

**Definition 9** (*POI-POI graph*) is a bidirectional bipartite graph pointing the spatial proximity between two locations. In particular, two locations  $l$  and  $l'$  are connected with an edge, if and only if one user  $u$  check-ins both of them during a time period  $t$  and a distance range  $R_g$ . According to this assumption we build our graph denoted as  $\mathcal{G}_{ll'} = (L \cup L, \mathcal{W}_{ll'})$ , where  $L$  is the set of locations, and  $\mathcal{W}_{ll'}$  is the set of the weighted edges between two location nodes considering their geographical proximity. This weight is computed as:

$$\mathcal{W}_{ll'} = 1 - \frac{(geodist_{l,l'})}{R_g}$$

where  $geodist_{l,l'}$  computes the geographical distance between two locations.

**Definition 10** (*POI-User graph*) is a directed bipartite graph which points the relation between a location and a user. The main difference in that graph is the weighting strategy we follow, since the influence of a location to a user differs from the previous approach. In particular, this graph is denoted as  $\mathcal{G}_{lu} = (U \cup L, \mathcal{W}_{lu})$ , where  $L$  is the set of locations, and  $U$  is the set of users, and  $\mathcal{W}_{lu}$  is the set of the weighted edges among them. The weight is computed as the fraction of the number of times a location  $l$  is visited by a user  $u$  to the number of all users checked-in that location:

$$\mathcal{W}_{lu} = \frac{\sum_{\forall c_u \in l_i} c_u}{\sum_{\forall c_U \in l_i} c_U}$$

**Definition 11** (*POI-Time Period graph*) is a directed bipartite graph which indicates the importance of a location during a time period. This graph is denoted as  $\mathcal{G}_{lt} = (L \cup T, \mathcal{W}_{lt})$ , where  $L$  is the set of locations,  $T$  is the set of Time periods, and  $\mathcal{W}_{lt}$  is the set of the weighted edges among them. The weight is computed as the fraction of the number of the check-ins performed at a location  $l$  by all users, during a time period  $t_i$ , against the total number of check-ins during all time periods at the same location:

$$\mathcal{W}_{lt} = \frac{\sum_{\forall c_U \in t_i} |n_{lt}|}{\sum_{\forall c_U \in T} |n_{lt}|}$$

**Definition 12** (*POI-Stay Points graph*) is a bidirectional bipartite graph that describes the significance of some locations for a user. In daily schedule users follow some routes and spend some time in each location of that route sequence. The elapsed time of each location indicates the importance of this location to this particular user. Thus, the more time is spent on a location, the higher the importance of that location. Algorithm 2 is being used to extract these important locations denoted as ‘stay points’ ( $st_i$ ).

---

#### Algorithm 2: STAYPOINTS EXTRACTION

---

**Input:**  $R$ : Set of users' routes  
**Output:**  $SP$ : a list of stay points of each route

```

1 for ( $i=0$ ;  $i < p$ ;  $i++$ ) do // For every route  $r_i \in R$ 
2   for ( $j=1$ ;  $j < r[i]_{num} - 1$ ;  $i++$ ) do // For every location  $l_j \in r_i$ 
3     Duration[j]  $\leftarrow l_j.t_i - l_{j-1}.t_i$ ;
4     StayPoint  $\leftarrow l.max(Duration)$ ; // Find  $l_j$  with maximum elapsed time in  $r_i$ 
5     ST[i].add(StayPoint);
6 return ST
```

---

Then we construct the  $\mathcal{G}_{lst} = (L \cup ST, \mathcal{W}_{lst})$  graph, where  $L$  is the set of locations,  $ST$  is the set of stay points, and  $\mathcal{W}_{lst}$  is the set of the weighted edges among them. The weight is computed as the fraction of the number of times one location is considered as a stay point in a route, against the total number of times this location is considered as stay point in all routes:

$$\mathcal{W}_{lst} = \frac{\sum_{\forall st_i \in r} l}{\sum_{\forall st_i \in R} l}$$

**Problem Definition:** “Given a user  $u$ , a location  $l$  and a time instance  $t$  (expressed as  $Q(u, l, t)$ ), and the check-in history, predict the top- $N$  unvisited proximate POIs to that target user.”

## 4 Proposed approach

Next, we present RELIN, an optimized solution for jointly learning the graph embeddings of different information networks in the same latent space and we propose a unified framework for POI recommendations.

### 4.1 Learning embeddings of a bipartite graph

Nodes that are directly connected to an edge  $e_{i,j}$  and weight  $w_{i,j}$ , consist the **first-order proximity**, that is the local pairwise closeness between two nodes. On the other hand, nodes that share many connections but they are not directly related to an edge, they belong to the same neighborhood, thus, there are most likely to be similar to each other. These nodes consist the **second-order proximity**, that is the similarity between two unlinked nodes according to their network structure. To extract this kind of proximity on uni-partite graphs, the LINE model [21] learns the embeddings of large graphs into a low dimensional space. With this work, we extend this model to learn the embeddings over bipartite graph nodes. Moreover, our model can be applied into all kind of bipartite graphs i.e., directed/undirected, weighted/unweighted, or combinations of them.

Given two disjoint sets  $\mathcal{G} = (\mathcal{S}_A \cup \mathcal{S}_B, \mathcal{W})$ , the nodes in  $\mathcal{S}_A$  that share many connections with  $\mathcal{S}_B$  but are not directly connected to each other, are most likely to have the same distributions. The conditional probability of one node  $v_j \in \mathcal{S}_B$  is generated through node  $v_i \in \mathcal{S}_A$  such as:

$$p(v_j|v_i) = \frac{\exp(\vec{v}_j^T \cdot \vec{v}_i)}{\sum_{u_k \in \mathcal{S}_B} \exp(\vec{v}_k^T \cdot \vec{v}_i)} \quad (1)$$

where the embeddings vectors of vertices  $v_i$ , and  $v_j$  are represented as  $\vec{v}_i$  and  $\vec{v}_j$ , respectively. Thus, for each node  $v_i \in \mathcal{S}_A$ , Eq. (1) defines the conditional distribution  $p(\cdot|v_i)$  to all the corresponding nodes in the set  $\mathcal{S}_B$ . Then, for each edge there is a weight which implies the strength of this tie. To retain the proximity of the unlinked nodes in  $\mathcal{S}_A$ , we let the described conditional distribution to approximate the empirical distribution  $\hat{p}(\cdot|v_i) = \frac{w_{i,j}}{\sum w_{i,k}}$  with the following objective function:

$$O = \sum_{u_i \in \mathcal{S}_A} \lambda_i d(\hat{p}(\cdot|v_i), p(\cdot|v_i)) \quad (2)$$

where  $d(\cdot|\cdot)$  denotes the Kullback-Leibler divergence of the conditional and the empirical distributions, and  $\lambda_i$  is a regularization parameter to tune the significance of node  $v_i$ . For simplicity, we set this parameter equal to the out-degree of each node. Thus, Eq. (2) corresponds to the minimization of the following objective function:

$$O = - \sum_{e_{i,j} \in \mathcal{W}} w_{i,j} \log p(v_j|v_i) \quad (3)$$

The vectors  $\{\vec{v}_i\}_{i=1..S_A}$  and  $\{\vec{v}_j\}_{j=1..S_B}$  that minimize Eq. (3) are the low-rank nodes representations in  $\mathbb{R}^d$ .

**Table 2** Complexity analysis

Sample edge from alias table	$\mathcal{O}(1)$
Negative sampling optimization	$\mathcal{O}(N + 1)$
Overall complexity	$\mathcal{O}(N \cdot E)$

## 4.2 Model optimization through negative sampling

To avoid the calculation of the conditional probability  $p(\cdot|v_i)$  which needs the summation of the entire set of nodes in Eq. (3), we apply negative sampling (NEG) [19] over each edge. In particular, we use the noisy distribution of each edge individually to sample  $N$  negative edges as described in the following objective function:

$$\log \sigma(\vec{v}_j^T \cdot \vec{v}_i) + \sum_{h=1}^N W_{u_n \sim P_n(u)} [\log \sigma(-\vec{v}_n^T \cdot \vec{v}_i)] \quad (4)$$

where  $\sigma(x) = 1/1 + \exp(-x)$  is the sigmoid function with output values [0-1], and  $P_n(v \propto d_u^{3/4})$  is the noise distribution in which the negative samples are chosen with a unigram distribution empirically tuned in [19], such that each node occurrence in the set is independent of all other node occurrences. Thus, selecting a node as a negative sample is related to the out-degree in that power. To further improve the solution of Eq. (4), we apply ‘Hogwild’ algorithm [20] based on the asynchronous stochastic gradient descent (ASGD). In particular, each time an edge  $e_{i,j}$  is sampled, we calculate the gradient of node  $v_i$  over the corresponding embedding vector  $\vec{v}_i$  as follows:

$$\frac{\partial \mathcal{O}}{\partial \vec{v}_i} = w_{i,j} \cdot \frac{\partial \log p(v_j|v_i)}{\partial \vec{v}_i}$$

Notice that the gradient of node  $v_i$  is multiplied with the weight related to that edge. Thus, tuning the learning rate of the model, may cause problems due the variance of the weights. On the one hand, ‘**overfitting**’ may occur to the gradients with large weights, if large learning rate is chosen according to edges multiplied with small weights values. On the other hand, ‘**underfitting**’ may occur to the gradients with small weights, if small learning rate is chosen for edges multiplied with large weights values.

To balance the learning rate of our model, we adopt the sampling method presented in [21]. In particular, we sample a random edge  $e_{i,j} \in [0 - W_{sum}]$ , where  $W_{sum}$  denotes the sum of all weights in the particular network, and then we examine the interval in which the particular sampled edge falls into, i.e.,  $[\sum_{j=0}^{i-1} w_j, \sum_{j=0}^i w_j]$ . Finally, we draw a sampled edge using alias table according to [12] which eventually reduces the complexity to  $\mathcal{O}(1)$ . Table 2 presents the over all complexity of edge sampling optimization.

## 4.3 Joint learning of graph dynamics

Given the input bipartite graphs, the next step is to integrate them into our model. Graphs that correspond to users’ relations are: User-User, User-POI, User-Route, and User-Time Period. On the other hand, graphs that correspond to the location ties with other networks are: POI-POI, POI-User, POI-Important locations, and POI-Time period. We collectively integrate the embeddings of these graphs corresponding to users’ and POIs ties, by minimizing the

objective function:

$$O = \underbrace{O_{uu} + O_{ur} + O_{ut} + O_{ul}}_{\text{user networks}} + \underbrace{O_{ll} + O_{lst} + O_{lt} + O_{lu}}_{\text{POInetworks}} \quad (5)$$

where each of the above objective functions is computed as:

$$\begin{aligned} O_{uu} &= -\sum_{e_{i,j} \in W_{uu}} w_{i,j} \log p(u_i | v_j), & O_{ll} &= -\sum_{e_{i,j} \in W_{ll}} w_{i,j} \log p(l_i | l_j) \\ O_{ur} &= -\sum_{e_{i,j} \in W_{ur}} w_{i,j} \log p(u_i | r_j), & O_{lst} &= -\sum_{e_{i,j} \in W_{lst}} w_{i,j} \log p(l_i | st_j) \\ O_{ut} &= -\sum_{e_{i,j} \in W_{ut}} w_{i,j} \log p(u_i | t_j), & O_{lt} &= -\sum_{e_{i,j} \in W_{lt}} w_{i,j} \log p(l_i | t_j) \\ O_{ul} &= -\sum_{e_{i,j} \in W_{ul}} w_{i,j} \log p(u_i | l_j), & O_{lu} &= -\sum_{e_{i,j} \in W_{lu}} w_{i,j} \log p(l_i | u_j) \end{aligned}$$

To minimize the objective function presented in Eq. (5), first we merge together edges of all unipartite and bipartite graphs, and then, in each step we update the model by sampling a new edge. The probability of sampling an edge corresponds to the weight related to that edge. This way our model walks through the heterogeneous bipartite graphs with respect to the inner and the outer vertices of the graphs and the weight influence.

The training of the model is performed jointly and dynamically as shown in Algorithm 3. More specifically, after initializing the embeddings (Line 1), we update each of them according to the corresponding sample (Lines 3–11). For example, if an edge connecting a user and a location is chosen, then both the locations' embeddings and the users' embeddings will be updated accordingly. The algorithm is trained until the desired number of samples is reached and then it returns the updated embeddings (Line 12). The simultaneous training improves the algorithm's learning process since a user's selection is always a result of many factors.

---

### Algorithm 3: RELIN

---

**Input:**  $\mathcal{G}_{uu}, \mathcal{G}_{ul}, \mathcal{G}_{ur}, \mathcal{G}_{ut}, \mathcal{G}_{ll}, \mathcal{G}_{lu}, \mathcal{G}_{lr}, \mathcal{G}_{lt}, S, N$

**Output:**  $\bar{\mathbf{u}}$ : User embeddings,  $\bar{\mathbf{l}}$ : POI embeddings,

$\bar{\mathbf{r}}$ : Route embeddings,  $\bar{\mathbf{t}}$ : Time period embeddings

$\bar{\mathbf{st}}$ : Stay point embeddings

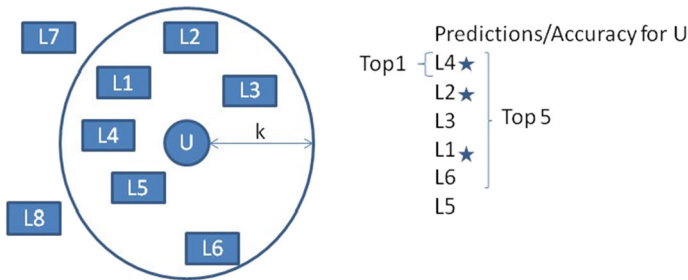
```

1 Initialize  $\bar{\mathbf{u}}, \bar{\mathbf{l}}, \bar{\mathbf{r}}, \bar{\mathbf{t}}, \bar{\mathbf{st}}$ ;
2 while (iteration  $\leq S$ ) do           // Until the desired number of samples reached
3   At each iteration draw  $N$  negative edges of the corresponding graph
4    $\mathcal{S}(w_{uu}) \in \mathcal{W}_{uu}$ , and update  $\bar{\mathbf{u}}$ ;
5    $\mathcal{S}(w_{ul}) \in \mathcal{W}_{ul}$ , and update  $\bar{\mathbf{u}}$  and  $\bar{\mathbf{l}}$ ;
6    $\mathcal{S}(w_{ll}) \in \mathcal{W}_{ll}$ , and update  $\bar{\mathbf{l}}$ ;
7    $\mathcal{S}(w_{lu}) \in \mathcal{W}_{lu}$ , and update  $\bar{\mathbf{l}}$  and  $\bar{\mathbf{u}}$ ;
8    $\mathcal{S}(w_{ut}) \in \mathcal{W}_{ut}$ , and update  $\bar{\mathbf{u}}$  and  $\bar{\mathbf{t}}$ ;
9    $\mathcal{S}(w_{lt}) \in \mathcal{W}_{lt}$ , and update  $\bar{\mathbf{l}}$  and  $\bar{\mathbf{t}}$ ;
10   $\mathcal{S}(w_{ur}) \in \mathcal{W}_{ur}$ , and update  $\bar{\mathbf{u}}$  and  $\bar{\mathbf{r}}$ ;
11   $\mathcal{S}(w_{lst}) \in \mathcal{W}_{lst}$ , and update  $\bar{\mathbf{l}}$  and  $\bar{\mathbf{st}}$ ;
12 return  $\bar{\mathbf{u}}, \bar{\mathbf{l}}, \bar{\mathbf{r}}, \bar{\mathbf{t}}, \bar{\mathbf{st}}$ 

```

---

In Fig. 3 we demonstrate an example, explaining the way RELIN will recommend locations to a user  $U$  over a range  $k$  where actual user preferences are marked by a star. After ranking all locations within the range, RELIN will recommend locations starting from the top. In this example, RELIN achieved perfect accuracy over Top@1 results, and 60% on Top@5.



**Fig. 3** An example of the recommendation process of RELINE

#### 4.4 Unified model for POI recommendations

By the time all embeddings, presented in previous section, have been learned, and given a prediction request  $Q(u, l, t)$ , concerning a user  $u$  in a location  $l$  at a timestamp  $t$ , we project these values to the corresponding time period  $t$ , route  $r$ , and stay point  $st$ , with geographical range distance less than 10 Km. We claim that one user is willing to attend proximate locations. Thus, given a recommendation further than this range, the probability of attending is very small. Then, we rank a list with the top@ $n$  unvisited candidate POIs for that user in that distance. The prediction score for each of the unvisited location is computed as:

$$Q(u, l, t) = \underbrace{\alpha \cdot (\vec{u}^T \cdot \vec{l})}_{\text{Social inf.}} + \underbrace{\beta \cdot (\vec{r}^T \cdot \vec{l})}_{\text{Geographical inf.}} + \underbrace{\gamma \cdot (\vec{t}^T \cdot \vec{l})}_{\text{Temporal inf.}} + \underbrace{\delta \cdot (\vec{st}^T \cdot \vec{l})}_{\text{Preference dynamics}} \quad (6)$$

where  $\vec{u}$  is the embedding of user  $u$ ,  $\vec{l}$  is the embedding of location  $l$ ,  $\vec{r}$  is the embedding of the route  $r$  this check-in belongs to,  $\vec{t}$  is the embeddings of the time period  $t$  in which the particular check-in was made, and  $\vec{st}$  is the embedding of the stay points. Moreover, RELINE learns jointly the embeddings from different information networks in the same latent space. Thus, the learned POI embeddings  $\vec{l}$  capture the information of all participated networks presented in the previous sections, such as  $uu$ ,  $ul$ ,  $ll$  etc. This way, we aim to eliminate sparsity, simply by using additional information networks. In particular, we jointly learn the dynamics of the *social influence* ( $\vec{u}^T \cdot \vec{l}$ ), the *geographical influence* ( $\vec{r}^T \cdot \vec{l}$ ), the *temporal influence* ( $\vec{t}^T \cdot \vec{l}$ ), and the *user's preference dynamics* ( $\vec{st}^T \cdot \vec{l}$ ), simultaneously. Finally,  $\alpha$ ,  $\beta$ ,  $\gamma$ , and  $\delta$  are regularization parameters that define the importance of each information network separately into our model. The regularization parameters are used to balance the effect the four different influences have on the final recommendation, due to the different nature and idiosyncrasies of users and the locations of the datasets.

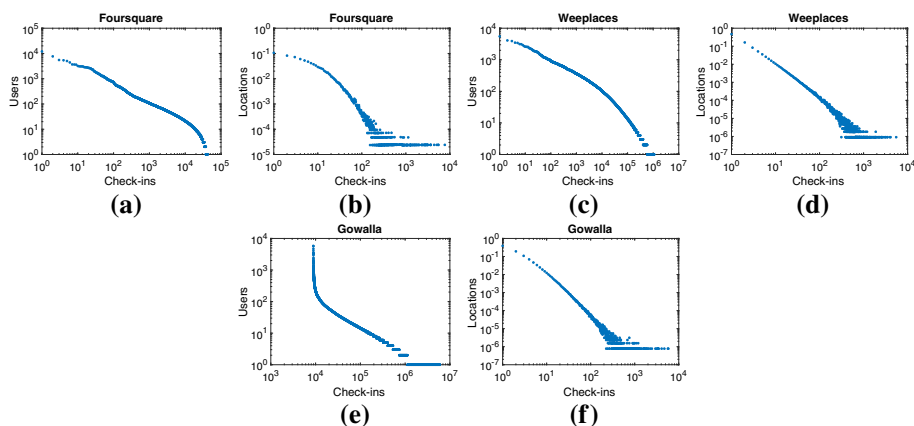
## 5 Performance evaluation

In this section, we focus on the performance of the proposed methodology. In particular, we compare our technique against previous ones. The source code of our approach is available at <https://github.com/thedx4/RELINe>.



**Table 3** Summary of datasets

	Foursquare	Weeplaces	Gowalla
Users	114,508	15,799	319,063
POIs	62,462	971,309	2,844,076
Check-ins	1,434,668	7,658,368	36,001,959
Friendships	32,511	119,930	1,900,653
Time span	Sep 2010–Jan 2011	Nov 2003–Jun 2011	Jan 2009–Aug 2011

**Fig. 4** Power law distribution diagrams for Foursquare (a) and (b), Weeplaces (c) and (d), and Gowalla (e) and (f)

## 5.1 Datasets and techniques

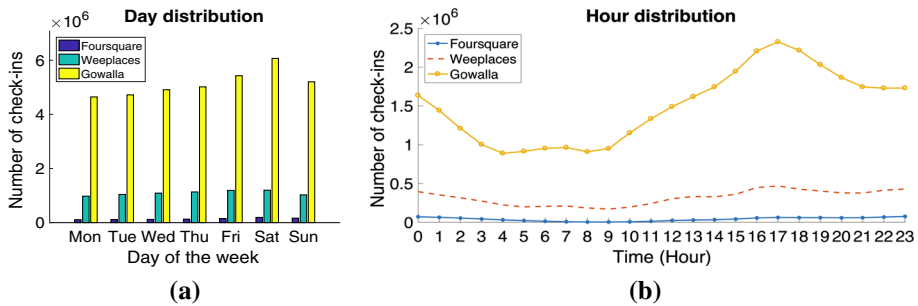
We have used three real-world datasets: (i) Foursquare [25]<sup>4</sup>, (ii) Weeplaces [1]<sup>5</sup>, and (iii) Gowalla [1]<sup>5</sup>. Their main characteristics are presented in Table 3. All datasets contain users' check-in history with timestamp and geographical information. Additionally, they contain information about the social ties of users' friendships. The datasets span a time period of 5, 91 and 31 months, respectively.

Figure 4 presents the distributions for all three datasets. As shown, the datasets follow a power law distribution for both the number of users' check-ins and the number of visits to a particular location. According to the power law distribution, there is a small number of users with many check-ins (short head) and many users that have a small check-in record (long tail). Similarly, the same principle holds for locations where, on the one hand, there are popular locations with a lot of visitors, and on the other hand, locations that are visited less frequently. Moreover, all datasets present a good example of the cold start problem [9], i.e., to recommend new locations to users with small history record.

Furthermore, we present the temporal distribution of all datasets in Fig. 5. In particular, Fig. 5a presents the distribution of the check-ins per day of the week. It is noticeable that users tend to be more active from Thursday to Sunday than the rest days of the week. Due to the fact that users check-in in locations during their spare time, this figure reflects a tendency

<sup>4</sup> <https://sites.google.com/site/dbhongzhi>.

<sup>5</sup> <http://www.yongliu.org/datasets>.



**Fig. 5** Check-in distribution per: **a** day of the week, and **b** hour of the day

according to which they use to perform more check-ins during the end of the week. The same principal stands for the distribution of the check-ins during the day as shown in Fig. 5b. We observe that users are more active during 13:00 p.m. to 2:00 a.m. with peak time at 18:00 p.m. and they tend to check-in at more locations during the night than during the morning.

Here we evaluate the performance of the following techniques:

- *Rank-GeoFM* [14] is a MF unified model which learns users' preferences and incorporates the spatial influence of proximate POIs simultaneously. The second term of the model is regularized with a distance-based weighting related to the target POI.
- *ASMF* [13] is another MF model consisted by a two step procedure. First, the model learns candidate POIs which have been visited by friends (social influence), and then a categorical-based weight is applied considering geographical influence.
- *GE* [25] is a Graph Factorization (GF) approach where two joint probability distributions are computed for each pair of nodes. The first is related to the adjacency matrix and the second is related to the embeddings. This method embed four bipartite information networks into the same latent space to predict the next unvisited POI by updating dynamically the users' profile.
- *PACE* [28] is a semi-supervised learning framework, based on users' Preference and Context Embeddings, that jointly learns users' and locations embeddings.
- *Versions of RELINE* To evaluate the influence of each network we have used three versions of our model. In particular, we start with a simplistic version and then we proceed with more enhanced alternatives.
  - **RELINE<sub>Social</sub>**: Is a simplified version of RELINE, which contains only the social influence information, that is  $(\bar{\mathbf{u}}^T \cdot \bar{\mathbf{I}})$ .
  - **RELINE<sub>Geographical</sub>**: Is the previous model enriched with the geographical influence  $(\bar{\mathbf{r}}^T \cdot \bar{\mathbf{I}})$ .
  - **RELINE<sub>Temporal</sub>**: Is the last version which enriches the former model with the Temporal influence  $(\bar{\mathbf{t}}^T \cdot \bar{\mathbf{I}})$ .

## 5.2 Evaluation methodology

We consider the partitioning of check-ins of each target user into three sets: (i) the training set  $\mathcal{E}_C^T$ , that is 80% of the total check-ins and is treated as known information, (ii) the probe set  $\mathcal{E}_C^P$ , that is 10% and is used for testing our model, and (iii) the validation set  $\mathcal{E}_C^V$  is the rest 10%

**Table 4** Impact of adding information networks to the model

Model	Acc@ <i>n</i>				
	1	5	10	15	20
<i>(a) Foursquare</i>					
RELINE <sub>Social</sub>	0.223	0.288	0.334	0.358	0.385
RELINE <sub>Geographical</sub>	0.257	0.291	0.342	0.377	0.407
RELINE <sub>Temporal</sub>	0.264	0.302	0.357	0.391	0.409
RELINE	<b>0.286</b>	<b>0.341</b>	<b>0.410</b>	<b>0.435</b>	<b>0.462</b>
<i>(b) Weeplaces</i>					
RELINE <sub>Social</sub>	0.315	0.349	0.406	0.420	0.445
RELINE <sub>Geographical</sub>	0.344	0.383	0.427	0.458	0.479
RELINE <sub>Temporal</sub>	0.331	0.377	0.413	0.445	0.463
RELINE	<b>0.386</b>	<b>0.421</b>	<b>0.488</b>	<b>0.514</b>	<b>0.536</b>
<i>(c) Gowalla</i>					
RELINE <sub>Social</sub>	0.363	0.405	0.505	0.518	0.526
RELINE <sub>Geographical</sub>	0.355	0.401	0.492	0.505	0.519
RELINE <sub>Temporal</sub>	0.347	0.387	0.473	0.493	0.508
RELINE	<b>0.408</b>	<b>0.447</b>	<b>0.518</b>	<b>0.533</b>	<b>0.556</b>

Bold represents the best result in accuracy

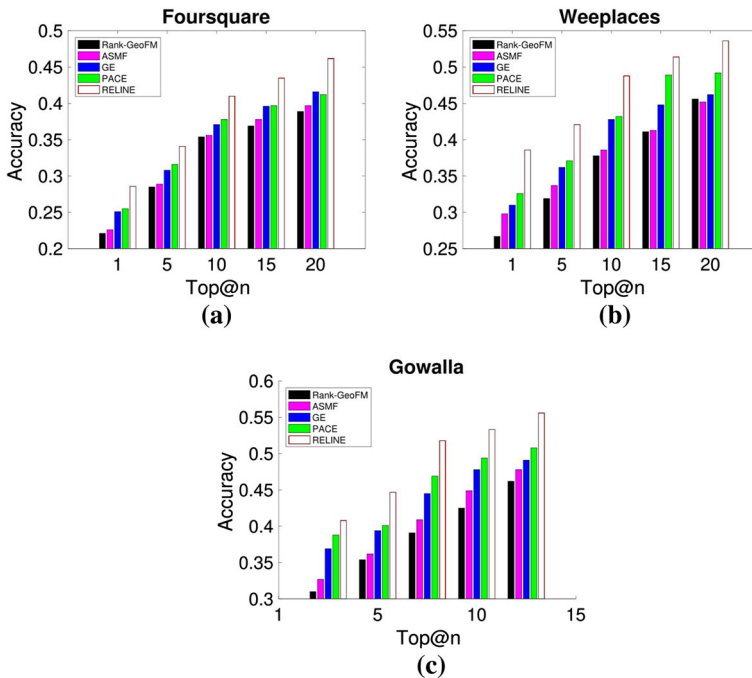
for tuning the hyper-parameters. It holds that,  $\mathcal{E}_C = \mathcal{E}_C^T \cup \mathcal{E}_C^P \cup \mathcal{E}_C^V$  and  $\mathcal{E}_C^T \cap \mathcal{E}_C^P \cap \mathcal{E}_C^V = \emptyset$ . Thus, for each target user we generate recommendations based only on the POIs in  $\mathcal{E}_C^T$ .

For the evaluation we measure the Accuracy@*n* as proposed in [24]. In particular, for each  $l \in \mathcal{E}_C^P$  given as a query  $Q(u, l, t)$ , we compute the prediction score for that  $l$  along with all unvisited proximate POIs of the target user with Eq. (6). We rank the predicted scores and then we select the top@*n* POIs. If the ground truth  $l \in \mathcal{E}_C^P$  appears in the top@*n*, then we have predicted correctly that location (i.e., true positive), otherwise our prediction is wrong. To compute the overall accuracy of the top@*n* we average all predictions test cases as:

$$\text{Accuracy@}n = \frac{\#\text{True Positive@}n}{\mathcal{E}_C^P} \quad (7)$$

### 5.3 Impact of information networks

Next, we examine the influence of each information network into the overall predictions accuracy. In particular, we explore how beneficial the enrichment is against the top-*n* predictions. The participant networks examined in this section are: (i) the *social influence*, (ii) the *geographical influence*, (iii) the *temporal influence*, and (iv) the *users' preference dynamics*. Thus, we compare RELINE with the three models RELINE<sub>Social</sub>, RELINE<sub>Geographical</sub>, and RELINE<sub>Temporal</sub>, which are described in Sect. 5.1. It is noticeable that as long as we embed more information networks, the accuracy increases as well, as shown in each of the columns of Table 4a–c. Thus, our model gradually alleviates the sparsity problem since it explores more information about the users or the POIs. Moreover, the accuracy of each model increases with *n*, meaning that the models fit well to users' behavior.



**Fig. 6** Accuracy versus the top@ $n$  recommendations for all users

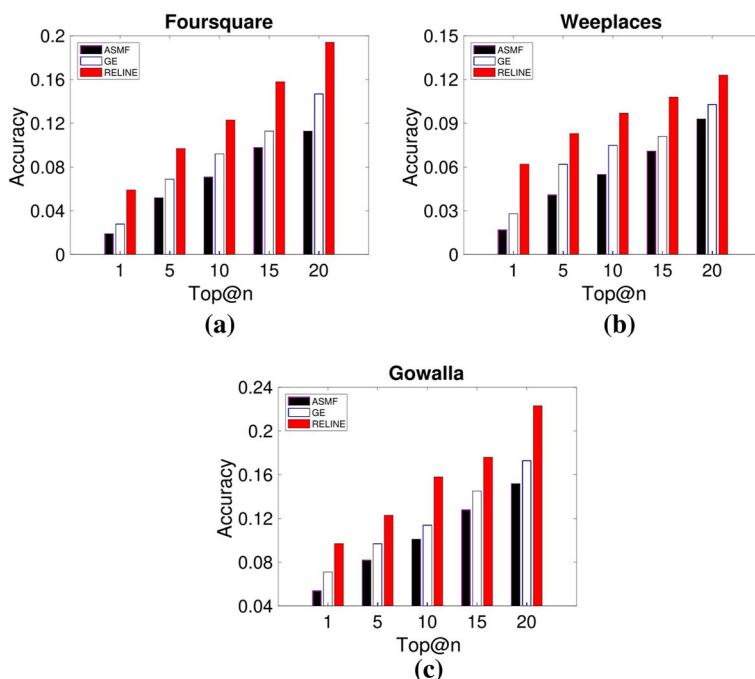
## 5.4 Comparison with other techniques

In this section we compare our model with other state-of-the-art approaches in terms of accuracy for the top- $n$  predictions [ $n = 1, 5, 10, 15, 20$ ] using three datasets. In particular, we examine the performance of all models for providing POI recommendations for: (i) all users, (ii) cold-start users, and (iii) cold-start POIs.

### 5.4.1 Accuracy over all users

First, we examine how models perform while providing recommendations to all users, without taking into account the history size or data sparsity. It is observed that RELINE significantly outperforms all methods as shown in Fig. 6.

Compared to methods that either: (i) learn users' preferences and then incorporate geographical influence like Rank-GeoFM, or (ii) learn social network check-ins like ASMF and ignore: (i) the POI sequence, (ii) the temporal influence, and (iii) preference dynamics, the performance is much higher since RELINE explores richer information in the same latent space. On the other hand, methods that learn users' and POI embeddings into the same latent space through multiple information networks, like GE and PACE, miss other important factors such as: (i) periodicity, (ii) preference evolution, and (iii) sequential importance of the check-ins and gain lower accuracy compared to our method. We notice that all models show higher accuracy when the number of POIs is small and the check-in activity denser, as shown in Fig. 6a. Moreover, in sparse datasets with multiple available POIs to visit, such as the other two datasets in Fig. 6b and c, the accuracy is lower. This finding supports the claim



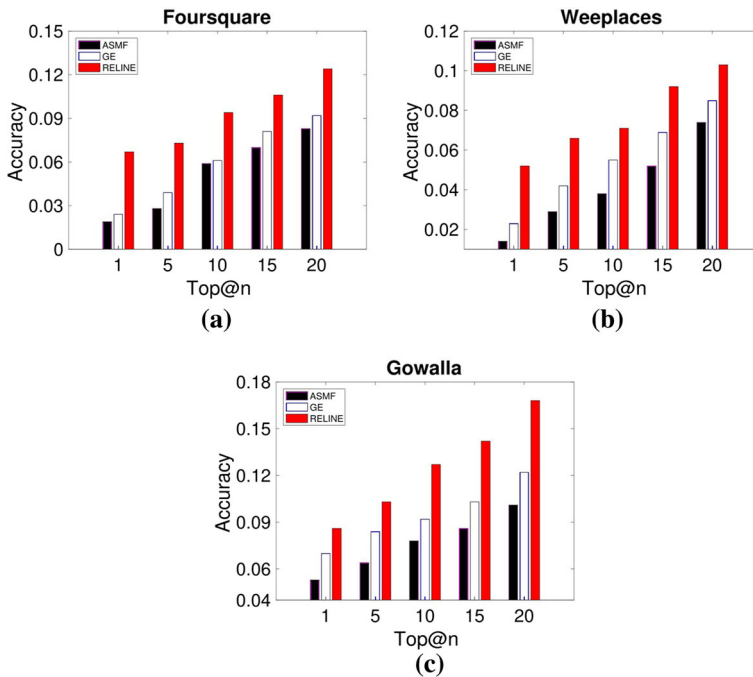
**Fig. 7** Accuracy versus the top@ $n$  recommendations for cold-start users

that learning both users' and POIs embeddings derived from as many information networks as possible, increases the model's ability to correlate a user with POIs and finally improves the accuracy.

### 5.4.2 Accuracy over cold-start users

Next, we examine the effectiveness of our model regarding the cold-start user and perform comparisons with other approaches. The concept was initially introduced in [9] and refers to users with short history. It is evident that supporting recommendations for such users is a difficult task due to lack of adequate information. In this context, we performed experiments providing recommendations only to cold-start users and we compared the performance of the models in terms of accuracy, as shown in Fig. 7.

Since only ASMF and GE support cold-start recommendations, we compare our model to these two methods. Moreover, we observe that the overall accuracy compared to the experiments of the previous section is reduced, which is normal, since we do not have much information. Even though, ASMF learns the POIs visited by target user's social network and then it refines the results with a categorical weighted strategy w.r.t. geographical influence, its accuracy is significantly lower compared to our method. Similarly, the performance of GE which jointly captures the spatial influence, the sequential effect, the periodicity, and the semantics into the same latent space is lower since it does not consider: (i) the importance of stay points in each sequence, (ii) the preference dynamics, (iii) the temporal effect, and (iv) the social influence. In contrast, we use side information related to both users and POIs from eight weighted information networks, which has a significant impact in effectiveness.



**Fig. 8** Accuracy versus the top@ $n$  recommendations for cold-start POIs

### 5.4.3 Accuracy over cold-start POIs

Next, we examine the problem of cold-start POIs. The goal in this case is to recommend unvisited POIs to users that have at least one check-in at a POI that has less than 15 check-ins. Thus, we examine not only how models behave on new users but also, how they do behave when a new location appears. In simple terms, we want to check if the new location is among the top- $n$  recommendations or not.

We evaluate our approach only with the models that support cold-start POI problem. In Fig. 8, we present the results of all models against all three datasets. Evidently, our model outperforms compared methods since, among all aforementioned factors, it explores POIs as a sequence of routes. In particular, all cold-start POIs are correlated with other proximate POIs when we extract users' routes. Also, some of them are considered as 'stay points' based on how long a user spends on that location. Then, we weigh the edge between the user and the cold-start POI, as an important one, if that user spends a lot of time there. This way, we tackle cold-start POIs by using their relational influence with other nodes on the graph during learning phase.

We note that both models gain lower accuracy for top-1 prediction when the number of the POIs is small as shown in Fig. 8a and b because they either explore users' friends check-ins who have not visited cold-start POIs, or they explore POIs sequences based on the frequency a user check-ins a cold-start POI. POIs with few check-ins are not connected to others in the POI-POI graph which GE uses. Finally, both models seem to gain higher accuracy with many POIs on the system as for example the Gowalla dataset which is pointed in Fig. 8c.



**Table 5** Impact of parameters  $S$  and  $d$  for top@10

S(mil)	$d$					
	70	80	90	100	110	120
<i>(a) Foursquare</i>						
50	0.383	0.385	0.385	0.388	0.388	0.388
100	0.392	0.398	0.390	0.391	0.391	0.392
<b>150</b>	0.405	0.408	0.409	<b>0.410</b>	0.411	0.411
200	0.405	0.408	0.408	0.409	0.411	0.411
250	0.423	0.408	0.408	0.409	0.411	0.411
<i>(b) Weeplaces</i>						
100	0.464	0.471	0.474	0.476	0.476	0.476
150	0.472	0.475	0.479	0.479	0.479	0.480
<b>200</b>	0.478	0.480	0.484	<b>0.488</b>	0.488	0.489
250	0.478	0.481	0.485	0.488	0.489	0.489
300	0.479	0.481	0.485	0.489	0.489	0.490
S(mil)	$d$					
	110	120	130	140	150	160
<i>(c) Gowalla</i>						
200	0.489	0.494	0.498	0.502	0.502	0.503
250	0.498	0.502	0.507	0.510	0.510	0.511
300	0.503	0.509	0.514	<b>0.518</b>	0.518	0.519
350	0.503	0.510	0.515	0.518	0.519	0.519
400	0.504	0.510	0.515	0.519	0.520	0.520

Bold represents the best result in accuracy

## 5.5 Parameter tuning

In this section, we study the importance of parameter tuning. In particular, we examine the impact of: (i) adding information networks to the model, (ii) the number of samples  $S$ , (iii) the embeddings dimensionality  $d$ , and (iv) the time period size  $T$ , to the performance of our model in terms of accuracy.

### 5.5.1 Impact of samples and dimensions number

Here, we present the experiments conducted to select the best candidate parameters for the number of samples and dimensions. The results for each dataset are presented in Table 5. Our findings for the top@10 indicate that our model is not greatly affected by the dimensionality  $d$ . The accuracy increases with higher rate along with dimensionality until  $d = 100$  for Foursquare and Weeplaces and  $d = 140$  for Gowalla. Thereafter, the accuracy does not change significantly. In contrast to dimensionality, our model is sensitive to the number of the samples ( $S$ ). Until a convergence point is reached, our model keeps increasing its accuracy along with the size of  $S$  and then the improvement is poor. Also, the higher the number of connections between the network edges, the higher the number of the sample. To gain higher accuracy, we set the number of samples equal to 100 for Foursquare, 200 for Weeplaces, and 300 for Gowalla along with the dimensionalities discussed previously.

**Table 6** Impact of time interval  $\Delta T$  over accuracy for the top@ $n$  recommendations

$\Delta T$	Acc@ $n$				
	1	5	10	15	20
<i>(a) Foursquare</i>					
5	0.254	0.317	0.381	0.396	0.438
10	0.261	0.325	0.397	0.403	0.446
15	0.273	0.328	0.402	0.425	0.453
<b>20</b>	<b>0.286</b>	<b>0.341</b>	<b>0.410</b>	<b>0.435</b>	<b>0.462</b>
25	0.273	0.335	0.402	0.420	0.453
30	0.265	0.328	0.397	0.416	0.446
<i>(b) Weeplaces</i>					
10	0.356	0.398	0.457	0.489	0.510
20	0.368	0.405	0.463	0.497	0.519
30	0.373	0.413	0.475	0.501	0.524
<b>40</b>	<b>0.386</b>	<b>0.421</b>	<b>0.488</b>	<b>0.514</b>	<b>0.536</b>
50	0.374	0.417	0.479	0.507	0.527
60	0.366	0.404	0.463	0.498	0.516
<i>(c) Gowalla</i>					
1	0.381	0.421	0.490	0.509	0.528
5	0.389	0.428	0.498	0.517	0.536
10	0.397	0.436	0.507	0.524	0.545
<b>15</b>	<b>0.408</b>	<b>0.447</b>	<b>0.518</b>	<b>0.533</b>	<b>0.556</b>
20	0.391	0.433	0.504	0.526	0.542
30	0.379	0.427	0.495	0.515	0.537

Bold represents the best result in accuracy

## 5.5.2 Impact of time period

Next, we examine the influence of the time interval  $\Delta T$  to the overall accuracy against varying values of the top@ $n$  predictions.  $\Delta T$  is crucial for our model since it is used to construct multiple graphs such as User-Route, User-Time period, POI-Time period etc. When extracting user-route edges, if there are not enough check-in data during small time intervals, the correlation of that user with other candidate POIs is difficult to be achieved. To overcome this issue, we use different size for  $\Delta T$  to examine which achieves higher accuracy. Table 6 presents the results for each dataset separately. It is noticeable that there is a point in each sub-table where the accuracy reaches its maximum value and then gradually decreases. The reason is that, when  $\Delta T$  is too small, there are few data and the accuracy is low. On the other hand, when  $\Delta T$  is large, there are too many nodes correlated to the target user which leads to overfitting. Thus, we set the size equal to 20, 40, and 15 for Foursquare, Weeplaces, and Gowalla, respectively.

## 5.5.3 Parameter tuning for $\alpha$ , $\beta$ , $\gamma$ , and $\delta$

Figure 9 presents the results of tuning the parameters  $\alpha$ ,  $\beta$ ,  $\gamma$ , and  $\delta$ . As shown in Eq. (6), each parameter corresponds to the social, geographical and temporal influences, along with the preference dynamics, respectively. For simplicity, we set each parameter  $p$  to a value

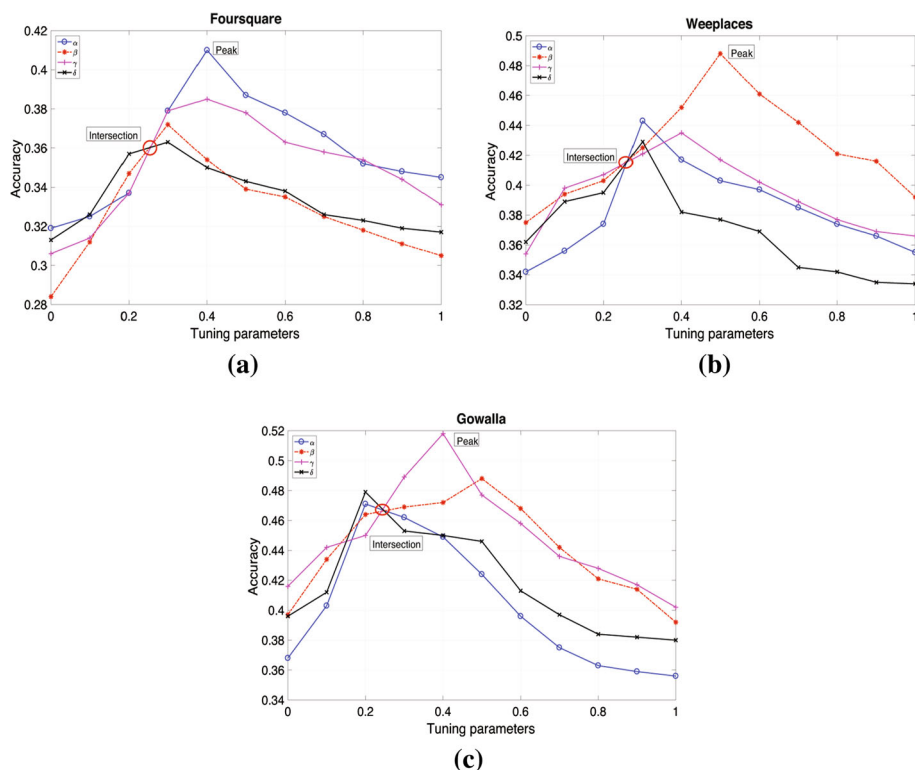


Fig. 9 Parameter tuning for all datasets

between  $[0-1]$  and all other parameters equal to  $(1 - p)/3$ . It is observed that there is an intersection point in each diagram where the accuracy meets. Also, there is a peak point where our model gets the higher accuracy, which was used to train our model. Moreover, the importance of each influential network is diversified trying to adapt the model to users' behavior.

As shown in Fig. 9, the influential parameter may be different. For example, regarding Foursquare, users are highly influenced by their social network as shown in Fig. 9a. Thus, they tend to visit locations visited by their friends. On the other hand, Weeplaces users tend to follow specific routes everyday, thus their movement to a new location is highly influenced by geographical factors, as shown in Fig. 9b. Finally, in datasets with many check-ins, like Gowalla, it is clear that capturing the preference dynamics is highly important for predicting the next location as Fig. 9c depicts.

## 6 Conclusions

The rapid growth of users' interaction in OSN and the huge amount of information they provide, led researchers to design new models that retrieve personalized information similar to past preferences. Unfortunately, these models are sensitive to sparsity and cold-start issues. The geographical information related to the content posted in such networks opened new

directions for research to attack both problems. There are many models that provide POI recommendations using either the social network influence, or the geographical proximity of user's current location. However, these models miss the temporal influence or users' preference dynamics, which are crucial while personalizing the retrieved information.

In this paper, we present a novel model that considers all aforementioned factors while providing POI recommendations. In particular, our model uses a probabilistic weighting strategy over multiple information graphs that correspond to users and POI relations. Then, it uses a graph-based approach that jointly learns users' and POI embeddings from these weighted graphs into the same latent space and provides personalized POI recommendations. We examine the influence of social, geographical, temporal, and preference dynamics in terms of accuracy. We compare our approach against four state-of-the-art models measuring the accuracy of the recommendations for: (i) all users, (ii) cold-start users, and (iii) cold-start locations. Our method significantly outperforms state-of-the-art approaches.

We identify two interesting directions for future work: (i) to incorporate node attributes into the model, and (ii) to enable incremental updates to the model to allow efficient adaptation in case of updates.

## References

1. Baral R, Li T (2016) MAPS: A multi aspect personalized POI recommender system. In: Proceedings 10th ACM conference on recommender systems (RecSys). Boston MA, pp 281–284
2. Chang S, Han W, Tang J, Qi G, Aggarwal CC, Huang TS (2015) Heterogeneous network embedding via deep architectures. In: Proceedings 21st ACM SIGKDD international conference on knowledge discovery and data mining (KDD). Australia, Sydney, pp 119–128
3. Chen H, Yin H, Wang W, Wang H, Nguyen QVH, Li X (2018) PME: projected metric embedding on heterogeneous networks for link prediction. In: Proceedings 24th ACM SIGKDD international conference on knowledge discovery and data mining (KDD). London, UK, pp 1177–1186
4. Cho E, Myers SA, Leskovec J (2011) Friendship and mobility: user movement in location-based social networks. In: Proceedings 17th ACM SIGKDD international conference on knowledge discovery and data mining (KDD). San Diego, CA, pp 1082–1090
5. Christoforidis G, Kefalas P, Papadopoulos A, Manolopoulos Y (2018) Recommendation of points-of-interest using graph embeddings. In: Proceedings 5th IEEE international conference on data science and advanced analytics (DSAA). Italy, Turin, pp 31–40
6. Gao H, Tang J, Hu X, Liu H (2015) Content-aware point of interest recommendation on location-based social networks. In: Proceedings 29th AAAI conference on artificial intelligence. Austin, TX, pp 1721–1727
7. He T, Chan KCC, Yang L (2018) Clustering in networks with multi-modality attributes. In: Proceedings IEEE/WIC/ACM international conference on web intelligence (WI). Santiago, Chile, pp 401–406
8. He T, Liu Y, Ko TH, Chan KCC, Ong Y (2020) Contextual correlation preserving multiview featured graph clustering. *IEEE Trans Cybern* 50(10):4318–4331
9. Herlocker JL, Konstan JA, Terveen LG, Riedl JT (2004) Evaluating collaborative filtering recommender systems. *ACM Trans Inf Syst* 22(1):5–53
10. Kefalas P, Manolopoulos Y (2017) A time-aware spatio-textual recommender system. *Expert Syst Appl* 78:396–406
11. Kumar V, Minz S (2016) Multi-view ensemble learning: an optimal feature set partitioning for high-dimensional data classification. *Knowl Inf Syst* 49(1):1–59
12. Li AQ, Ahmed A, Ravi S, Smola AJ (2014) Reducing the sampling complexity of topic models. In: Proceedings 20th ACM SIGKDD international conference on knowledge discovery and data mining (KDD), pp 891–900. New York, NY
13. Li H, Ge Y, Hong R, Zhu H (2016) Point-of-interest recommendations: learning potential check-ins from friends. In: Proceedings 22nd ACM SIGKDD international conference on knowledge discovery and data mining (KDD). San Francisco CA, pp 975–984

14. Li X, Cong G, Li XL, Pham TAN, Krishnaswamy S (2015) Rank-GeoFM: a ranking based geographical factorization method for point of interest recommendation. In: Proceedings 38th international ACM conference on research and development in information retrieval (SIGIR). Santiago, Chile, pp 433–442
15. Liu B, Qian T, Liu B, Hong L, You Z, Li Y (2017) Learning spatiotemporal-aware representation for POI recommendation. In: CoRR [arXiv:1704.08853](https://arxiv.org/abs/1704.08853)
16. Liu L, Xu J, Liao SS, Chen H (2014) A real-time personalized route recommendation system for self-drive tourists based on vehicle to vehicle communication. *Expert Syst Appl* 41(7):3409–3417
17. Lu Z, Savas B, Tang W, Dhillon IS (2010) Supervised link prediction using multiple sources. In: Proceedings 10th IEEE international conference on data mining (ICDM). Australia, Sydney, pp 923–928
18. Meng F, Gao D, Li W, Sun X, Hou Y (2013) A unified graph model for personalized query-oriented reference paper recommendation. In: Proceedings 22nd ACM international conference on information and knowledge management (CIKM). San Francisco CA, pp 1509–1512
19. Mikolov T, Sutskever I, Chen K, Corrado G, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: Proceedings 26th international conference on neural information processing systems (NIPS). Lake Tahoe, NV, pp 3111–3119
20. Niu F, Recht B, Re C, Wright SJ (2011) HOGWILD!: a lock-free approach to parallelizing stochastic gradient descent. In: Proceedings 24th international conference on neural information processing systems (NIPS). Granada, Spain, pp 693–701
21. Tang J, Qu M, Wang M, Zhang M, Yan J, Mei Q (2015) LINE: large-scale information network embedding. In: Proceedings 24th international conference on world wide web (WWW). Florence, Italy, pp 1067–1077
22. Vasuki V, Natarajan N, Lu Z, Savas B, Dhillon I (2011) Scalable affiliation recommendation using auxiliary networks. *ACM Trans Intell Syst Technol* 3(1):1–20
23. Wang H, Fu Y, Wang Q, Yin H, Du C, Xiong H (2017) A location-sentiment-aware recommender system for both home-town and out-of-town users. In: Proceedings 23rd ACM SIGKDD international conference on knowledge discovery and data mining (KDD). Halifax, Canada
24. Wang W, Yin H, Sadiq S, Chen L, Xie M, Zhou X (2016) SPORE: A sequential personalized spatial item recommender system. In: Proceedings 32nd IEEE international conference on data engineering (ICDE). Helsinki, Finland, pp 954–965
25. Xie M, Yin H, Wan H, Xu F, Chen W, Wang S (2016) Learning graph-based POI embedding for location-based recommendation. In: Proceedings 25th ACM international on conference on information and knowledge management (CIKM). Indianapolis, IN, pp 15–24
26. Xiong L, Chen X, Huang TK, Schneider J, Carbonell JG (2010) Temporal collaborative filtering with Bayesian probabilistic tensor factorization. In: Proceedings 10th SIAM international conference on data mining (SDM). Columbus, OH, pp 211–222
27. Xu C, Tao D, Xu C (2013) A survey on multi-view learning. In: CoRR [arXiv:1304.5634](https://arxiv.org/abs/1304.5634)
28. Yang C, Bai L, Zhang C, Yuan Q, Han J (2017) Bridging collaborative filtering and semi-supervised learning: A neural approach for POI recommendation. In: Proceedings 23rd ACM SIGKDD international conference on knowledge discovery and data mining (KDD). Halifax, Canada, pp 1245–1254
29. Yin H, Wang W, Wang H, Chen L, Zhou X (2017) Spatial-aware hierarchical collaborative deep learning for POI recommendation. *IEEE Trans Knowl Data Eng* 29:2537–2551
30. Yuan Q, Cong G, Ma Z, Sun A, Thalmann NM (2013) Time-aware point-of-interest recommendation. In: Proceedings 36th ACM international conference on research and development in information retrieval (SIGIR). Dublin, Ireland, pp 363–372
31. Zhang JD, Chow CY, Li Y (2014) LORE: Exploiting sequential influence for location recommendations. In: Proceedings 22nd ACM SIGSPATIAL international conference on advances in geographic information system (GIS). Dallas, TX, pp 103–112
32. Zhao S, Zhao T, King I, Lyu MR (2017) Geo-teaser. In: Companion proceedings 26th international conference on world wide web (WWW). Perth, Australia, pp 153–162



**Giannis Christoforidis** received his 5-year Diploma degree in Informatics and Telecommunications Engineering from University of Western Macedonia (2014) and his M.Sc. degree in Networking and Web Development from Aristotle University of Thessaloniki (2018). Currently, he is a Ph.D. candidate at the same University and his research interests are based on knowledge discovery from big data.



**Pavlos Kefalas** received his B.Sc. degree in Computer Science (2009) from University of Crete, and his M.Sc. degree in Information Systems from Aristotle University of Thessaloniki (2011). He received his Ph.D. degree in recommendation models for dynamic spatio-temporal big data from the same university (2017). His research interests include recommender systems, reasoning, information retrieval, machine learning, distributed databases, spatio-temporal and graph mining.



**Apostolos N. Papadopoulos** received his 5-year Diploma degree in Computer Engineering and Informatics from University of Patras (1994) and his Ph.D. degree from Aristotle University of Thessaloniki (2000). His research interests include databases, data mining and big data analytics. The paper entitled “SkyGraph: An Algorithm for Important Subgraph Discovery, received the award for the best Knowledge Discovery paper in ECML/PKDD 2008. Moreover, the paper “Metric-Based Top- $k$  Dominating Queries” that was presented in EDBT 2014, has been selected as the best paper and an extended version appears in ACM Trans. on Database Systems. Currently, he is an Associate Professor at the Department of Informatics of the Aristotle University of Thessaloniki.





**Yannis Manolopoulos** is a Professor and Vice-rector with the Open University of Cyprus. In the past, he has been with the University of Toronto, the University of Maryland at College Park, the University of Cyprus and the Aristotle University of Thessaloniki. He has served as Rector of the University of Western Macedonia, Greece and Vice-Chair of the Greek Computer Society. He has co-authored 5 monographs published by Springer, 8 textbooks in Greek, as well as >300 journal and conference papers related to Data Management. He received >13000 citations from >2000 distinct academic institutions and 5 best paper awards from SIGMOD, ECML/PKDD and MEDES (2) and ISSPIT conferences. He has also served as main co-organizer of several major fora, among others: ADBIS'2002, SSTD'2003, SSDBM'2004, ICEIS'2006, EANN'2007, ICANN'2010, AIAI'2012, WISE'2013, CAISE'2014, MEDI'2015, ICCCI' 2016, TPD'2017, DAMDID'2017, DASFAA'2018, EAIS'2018, WIMS' 2018, IDEAS'2019, MEDES'2019 conferences. He has acted as eval-

uator for funding agencies in Austria, Canada, Cyprus, Czech Republic, Estonia, EU, Georgia, Greece, Hong-Kong, Israel, Italy, Lithuania, Poland and Russia. Currently, he serves in the Editorial Board of Information Systems Journal, The World Wide Web Journal, The Computer Journal, among others.