IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 5, 2019, accepted April 30, 2019, date of publication May 8, 2019, date of current version May 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2915535

Motion Based Inference of Social Circles via Self-Attention and Contextualized Embedding

TING ZHONG[1], (Member, IEEE), FANG LIU[1], FAN ZHOU[1], (Member, IEEE), GOCE TRAJCEVSKI[2], (Member, IEEE), AND KUNPENG ZHANG[3], (Member, IEEE)

[1]School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China
[2]Department of Electrical and Computer Engineering, Iowa State University, Ames, Iowa 50011, USA
[3]Department of Decision, Operations & Information Technologies, University of Maryland, College Park, MD 20742, USA

Corresponding author: Fan Zhou (fan.zhou@uestc.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61602097 and Grant 61472064, in part by the National Science Foundation (NSF) Grant III 1213038 and Grant CNS 1646107, and in part by the Office of Naval Research (ONR) Grant N00014-14-10215.

ABSTRACT Extracting knowledge from human mobility data is an important task for many downstream applications such as point-of-interest recommendation, motion trace identification, and personalized trip planning. A specific problem that has recently spurred research interest is the so-called Social Circle Inference from Mobility data (SCIM), aiming at inferring relationships among users based on mobility data and without any explicit structured network information. The existing methods either require partial social ties or fail to model the implicit correlations between user links, thereby suffering from critical inference bias. We present a novel SCIM framework, called SCIM via self-Attention and Contextualized-embedding (SCIMAC) – a methodology capturing multiple aspects of users' check-in behavior and complex motion patterns of different users. Instead of directly applying the recurrent model on training user trajectories, the proposed method introduces a new module for context-aware check-in representation learning by adaptively incorporating the internal states of the recurrent layers, which is more effective than the context-independent check-in embedding used in existing social circle inference approaches. To model the underlying correlations between labels, SCIMAC leverages a more sophisticated label embedding technique to adjust the penalties for correlated users, enabling a better understanding of the user's hierarchy in the label space, and alleviating the inference bias. We empirically demonstrate that our SCIMAC model significantly outperforms several state-of-the-art baselines on real-world datasets.

INDEX TERMS Social circle inference, self-attention, contextualized embedding, mobility learning, multi-label classification.

I. INTRODUCTION

The past decade has witnessed a rapid growth of both academic and practical interest on mining human mobility patterns from location based social networks (LBSN) such as Twitter, Foursquare and Weibo. Availability of large volumes of LBSN data has spurred research in studying user behavior and movement patterns that can be analyzed for various LBSN services such as point-of-interest (POI) recommendation [1], tour scheduling [2], associating users with specific trajectories [3], [4], etc.

In addition to the geographical footprint, users in LBSN also interact with each other in a virtual world, where the social connections are usually depicted as the follower/followee relationships. One of the key applications in social networks is identifying the links between users – commonly referred to as a link prediction problem [5] – either inferring the links that are likely to occur in the near future or reconstructing the existing links that are missing in the current snapshot of the social network. Link prediction can benefit many downstream services such as item/friend recommendation [6], [7], scientific recommendation [8], identifying an online community [9] or predicting possible missing links between suspects in an organized crime [10].

The most common approaches for inferring links between the nodes are based on some information about the network structures [11]–[15]. However, assuming the availability of such information may not be practical in many applications settings, since the structural information is too sensitive to be shared with third-party service providers. Recent studies have demonstrated that the social relationships can be unveiled by mining users' mobility data [16]–[20], which has been

The associate editor coordinating the review of this manuscript and approving it for publication was Theofanis P. Raptis.

cast as a Social Circle Inference from Mobility data problem (SCIM). Using geographical data to infer the underlying social relations between mobile users is of interest to many commercial applications such as item recommendation and suspect identification. While such approaches have demonstrated advantages in inferring links from mobility data, more often than not they also require some prior knowledge of the network structures. For example, vec2link [19] unified users' offline check-in behavior and their online structures for improving the link prediction performance, where the social relations are exploited by network representation techniques [21]. Recent work [20] proposed the similar solutions except that it only utilized a smaller (sub)set of the social ties as the observation of the network.

Complementary to these, walk2friends [22] and TSCI [23] considered variants of the SCIM problem, i.e., inferring the social relations *solely* from the mobility data. The work in [22] addressed this problem from the perspective of privacy protection, where an inference attack by learning user's mobility features is derived. However, the proposed attack approach simply relies on the embedding of users' check-in histories with word2vec [24] technique – therefore, its efficiency is questionable. TSCI [23] presented a distinct variant of the problem – inferring the social ties based on a trajectory of an anonymous user/generator (rather than knowing the user himself). The inference was formulated as a multi-label classification problem and the solution presented relied on a variational auto-encoder [25].

While achieving advantages in social ties prediction, existing methods suffer from the problem of "inference bias" originated from predicting the links separately, i.e., ignoring the fact that many users may be implicitly correlated. From a complementary perspective, different parts of a particular user's trajectory may have different impacts on different links, thereby having varying influence on correct links prediction. In addition to not considering the different impacts of sub-trajectories, the existing approaches that incorporate mobility usually model the trajectories with recurrent neural network (RNN) based models such as LSTM [26] or GRU [27], combined with POI embedding via pre-training methods such as word2vec. In turn, the typical representation of the POIs is with fixed vector, thus failing to capture the complex characteristics of user check-in behavior and the different aspects of POIs. For instance, the check-in of a particular restaurant may appear in many users' footprints, which may have different meanings for different individuals – e.g., it may meet the taste of a particular food lover; however, it may also mean that the restaurant's location is near to the user's place of residence.

In this paper, we propose a novel social circle inference framework SCIMAC (SCIM via self-Attention and Contextualized-embedding) that can overcome the limitations for implicit link prediction – focusing on scenarios where mobility data is available, but there is no explicit data regarding the social relations. Specifically, we present a new POI embedding model that learns all the internal states of the trajectory LSTM model and captures context-aware aspects of check-in meanings, which can be directly used for downstream LBSN services. To effectively capture the implicit user correlations without access to any network structures, we propose a label-correlated classification method, represented by a label embedding layer operating in the latent label space. We also propose to calculate the importance of each POI in a user's check-in trajectory with a self-attention module, which can be seamlessly coupled and jointly updated with the encoder-decoder models, so as to reflect the user preference on check-ins in multiple aspects. In summary, our main contributions are:

- We address the SCIM problem in LBSN applications as a novel learning paradigm by analyzing human mobility patterns. We propose an approach for encoding the semantics of trajectories and inferring the trajectory context, which is the first context-aware trajectory embedding model distinguishing the semantics of POI embedding in different trajectories, and opens a new perspective for understanding user check-in behavior.
- We exploit the self-attention mechanism to refine the representation of a sequence of check-ins against itself and to better learn various aspects of users' preference over POIs, which can efficiently model the dependencies and importance of user long-short term motion patterns.
- We introduce a novel label embedding method by considering the implicit correlations between users when inferring the circles, which can largely alleviate the inference bias in existing multi-label classification based approaches and improve the link prediction accuracy.
- To demonstrate the effectiveness of SCIMAC, we conducted extensive experiments on several real-world datasets. The results show that SCIMAC can both improve the social circle inference accuracy compared to the state-of-the-art approaches, and also explain its behavior.

We note that our earlier work [23] addressed the trajectory-based inference of social connections, however, the present article is substantially different in several aspects:

1) We have improvised the incorporation of the semantics of the POIs in the learning, by including a self-attention layer to extract the multi-aspect preference of users over POIs.
2) We introduce a novel pre-training procedure and labelling methodology.
3) We propose a novel, more sophisticated architecture, which enables incorporation of implicit user correlations in the embedded latent space.
4) We provide novel experimental evaluation, comparing the benefits of SCIMAC against the TSCI method from [23] (in addition to other approaches).

The remainder of this article is organized as follows. We review the relevant related works in Section II and introduce the problem definition in Section III. The details of our SCIMAC model are presented in Section IV. Experimental results demonstrating the superiority of our model are discussed in Section V, followed by conclusions and directions for future work in Section VI.

II. RELATED WORK

We now review the relevant literatures in social circle inference and human mobility mining, and position our work in the context of the existing results.

A. SOCIAL CIRCLE INFERENCE

The link prediction problem [5] targets the identification of missing links, or links that are likely to be formed in the future given the current social network, and is at the core of many applications, a prominent example of which is "People You May Know" in online social networks such as Twitter, Facebook, Weibo, LinkedIn, etc. Link prediction has attracted significant attentions from both industry and academia based researchers over the last decade. Recent advances in network representation learning [11]–[13], [28] and graph neural networks [14], [15], [29], [30] have facilitated the link prediction task from the perspective of network embedding and structure learning, where the propensity of forming a link is based on the nodes similarity.

However, accessing social network structures may not be feasible in practice due to privacy of the sensitive data possessed by service providers. Often, researchers turn to inferring the social interactions[1] from various kinds of auxiliary information such as communication/call data [32] and user opinions [33]. Concurrently, as increased amount of mobility data becomes available due to the development of location technologies, there is a body of works focusing on inferring social relationships from user offline mobility data [16], [20], [22], [34], [35]. As demonstrated in [17]–[19], [22], mobility data can indeed serve as a strong predictor for inferring social ties.

For example, informative features capturing local and global spatio-temporal factors of trajectories were used in [36] to infer two users' social relationship. walk2friends [22] is a word2vec [24] based model that relies on user mobility features for inferring social links, although it focuses on the security issues of social inference attack and proposes defense mechanisms against privacy risks stemming from mobility data sharing. Matrix factorization, a well-established technique normally used in recommender systems, has also been successfully applied in link prediction and friend recommendation [37], [38], and inferring social network structures in ecology [39]. In a similar spirit, vec2Link [19] is a hybrid link prediction framework that captures user offline location preference and online social preference, respectively learned from user spatial activities and network representations. In this vein, a most recent work O2O-Inf [20] proposed the model for inferring the social ties based on a small set of observed social links and features characterizing geographical interactions between nodes.

[1] This problem is synonimously referred to as social link inference [22], link recommendation [31], social circle inference [23] and social ties inference [20] in the literature.

The significant progress on social circle/ties inference from user trajectories enabled by the previous methods, has the drawback that the proposed solutions either (partially) rely on social network structures [20], or require hand-craft feature engineering on modeling effective user co-location features from their footprints [22]. To alleviate this, in our previous work [23] we addressed a novel problem – *trajectory-based social circle inference (TSCI)* – which predicts the social circle solely from user check-in behaviors in an end-to-end manner and captures the motion patterns with the Bayesian generative networks. We re-iterate that the model proposed in this work is a significant modification and extension of TSCI, with three major distinctions: (1) SCIMAC adds a self-attention layer to extract the multi-aspect preference of users over POIs compared to the simple word2vec based check-in embedding used in TSCI and the previous works such as walk2friends [22]; (2) SCIMAC simplifies the trajectory semantics learning in TSCI and substitutes with a novel contextualized pre-training procedure to learn the context-dependent trajectory representations; (3) Most importantly, SCIMAC improves the inference performance by considering the implicit user correlations when inferring the circles, where the inference is performed in the embedded latent label space in a unified manner, rather than the respective prediction in TSCI – which is prone to inconsistent inference and, as we will show, is inferior in performance.

B. HUMAN MOBILITY MINING

One of the core aspects of the model proposed in this work is the uncovering of semantic patterns characterizing human trajectory. Several aspects of related (variants of the) problems have been studied extensively and, for the purpose of comparing with our findings, the results can be categorized into:

(1) *statistical patterns learning*: measuring and quantifying models such as continuous-time random-walk [40] and Lévy flight [41], or accounting for characteristics of individual human trajectories [42].

(2) *similarity mining*: detecting mobility similarity and capturing moving patterns [43], [44], along with exploiting trajectory semantics [38], [45].

(3) *periodical pattern mining*: finding (sub-)sequences and periodical motion patterns, enabling travel recommendation [2], life pattern understanding [35], recovering trajectories associated with users [3], [4] and next location prediction [46], [47].

Recently, deep learning techniques – especially ones based on recurrent neural networks (RNNs) such as Long-short Term Memory (LSTM) [26] and Gated Recurrent Units (GRU) [27] – have been widely used to capture the long term sequential influence and mobility patterns. Spatial-temporal RNN models [3], [46]–[48] extend the RNN model by incorporating temporal and spatial context in each time unit for various downstream tasks, such as trajectory classification [3], [4], POI recommendation [1], [49], [50] and prediction [46], [47]. However, these methods mostly focus on capturing the transition dependencies among POIs – and they neither explicitly model users' mobility similarity nor infer their social interactions. In addition, the existing deep mobility learning models usually embed POIs with low-dimensional vectors learned from word2vec and use the last hidden state of RNN to represent the trajectory during training. This, however, allows only a single context-independent representation of both POIs and trajectories. In this regard, SCIMAC is the first spatio-temporal learning model that considers the context-dependent features and intermediate interactions of layers for specific tasks such as social circle inference.

III. PRELIMINARIES

We now proceed with introducing the basic terminology and formalizing the problem in the context of social ties inference.

A POI is defined as a location of relevance obtained, for example, as a GPS value. It can correspond to a centroid of a region; an address of an object, etc. – which can be uniquely identified in a suitable coordinate system. Let $\Gamma = \{T_1, T_2, \cdots, T_M\}$ denote the set of all the trajectories (corresponding to the users) and let $T_i = \{c_1^i, \cdots, c_j^i, \cdots, c_N^i\}$ denote a trajectory generated by the i-th user, where $c_j^i \in \mathbf{C}$ ($j \in [1, N]$) is the j^{th} check-in for this user. Whenever there is no ambiguity, we will omit the user's index in the superscript.

Table 1 summarizes the notations used in this article.

TABLE 1. Notations.

Notation	Description
u_i	a user (label).
$\mathbf{u}_i$	the distribution vector of a label.
$c_i/\mathbf{c}_i$	check-in and the embedding vector.
$\mathbf{C}$	the set of check-ins.
T_i	a trajectory.
M	the number of all trajectories.
N	the number of check-ins in a trajectory.
$\mathbf{h}_i$	hidden state of encoder for pre-training.
$\tilde{\mathbf{h}}_i/\mathbf{s}_t$	hidden state of encoder and decoder.
$\hat{\mathbf{z}}_i$	the weighted embedding vector of c_i.
$\mathbf{z}_i$	the final embedding vector of c_i in ELMo.
$\mathbf{a}_t$	the context vector at time step t.
$\mathbf{M}_t$	the mask vector at time step t.

We assume that a social network is represented as an undirected graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ where $\mathbf{V}$ is the set of nodes corresponding to the individual users, and $\mathbf{E}$ is the set of edges connecting the users. Without loss of generality, for the purpose of this work we assume that the edges are unweighted (i.e., each edge has a weight 1).

A. SOCIAL CIRCLE INFERENCE (SCI) PROBLEM

The goal of the SCI is to reconstruct the social circle of a particular user u_i – i.e., to determine (and/or predict) the links between a set of users $\mathbf{U}$ ($\mathbf{U} \subseteq \mathbf{V}$) and the given user u_i.

A variant of the SCI that is at the core motivation for this paper is the *Social Circle Inference from Mobility (SCIM)* problem in which we are given a trajectory T_i and the objective is to infer the unobserved social ties in $\mathbf{G}$ based on a few observations of historical check-in trajectories of all the users in the network. While the problem of SCIM has been studied from various aspects, following [23] in this work we focus on two specific tasks of SCIM:

B. SCIM TASK I (SCIM-I)

Given a trajectory T_i generated by a known user u_i, SCIM-I learns a model $\mathcal{M}$ to identify a set of users $\mathbf{U}_i$ ($\mathbf{U}_i \subseteq \mathbf{V}$) who are in the same social circle of u_i or u_i's friends: $\mathcal{M}(T_i) \mapsto \mathbf{U}_i$.

In SCIM-I, we use a subset of trajectories of each user to learn the latent patterns of mobility and predict the social ties based on the offline geographical activity.

The second task that we address can be specified as:

C. SCIM TASK II (SCIM-II)

Unlike SCIM-I, in SCIM-II we do not know who generated the trajectory T_i.

More specifically, the user who generated T_i in SCIM-II is anonymous and his/her corresponding trajectories may never appear in the training set. This variant is also known as prediction of *the circle of a trajectory* [23] and is suitable for tackling the cold-start problem for new users.

We formulate the two types of SCIM as the multi-label classification problem [51], and denote the label space as $\mathbf{V} = (u_1, u_2, ..., u_{|\mathbf{V}|})$, where $|\mathbf{V}|$ refers to the total number of all users in the network. Obviously, each trajectory T_i is associated with a subset $\mathbf{U}_i$ of $\mathbf{V}$ (i.e., $\mathbf{U}_i \subseteq \mathbf{V}$). Therefore, the two tasks of SCIM are to learn classifiers that link trajectories to members who are friends of their owner: $T_i (\in \Gamma) \mapsto \mathbf{U}_i (\subseteq \mathbf{V})$.

IV. SCIMAC: ARCHITECTURE AND PROCESSING

The overall framework of the proposed model SCIMAC is shown in Figure 1. In particular, it consists of three main components: (1) contextualized POI embedding; (2) trajectory modeling with recurrent model and self-attention; and (3) label embedding and sequential circle member inference.

With this model as a reference, in the rest of this section we first discuss how the data is pre-processed, followed by the proposed approach for POI representation in a manner that will preserve both the sequentiality and the semantics of POI visits. We conclude this section with a detailed discussion of how we infer social circles from trajectories, while avoiding inference bias.

A. DATA PRE-PROCESSING

Recall that each trajectory T_i generated by a user u_i, is represented as a sequence of consecutive check-ins – and we assume that there is a time-instant associated with each check-in, i.e., $T_i = (c_{1,t_1}^i, c_{2,t_2}^i, \cdots, c_{n,t_n}^i)$, where c_{j,t_j}^i is j-th POI visited by user u_i at time t_j. A given trajectory T can be segmented as $T = (T^1, T^2, \cdots, T^m)$, meaning that there are m sub-trajectories within the time interval $[t_1, t_n]$,

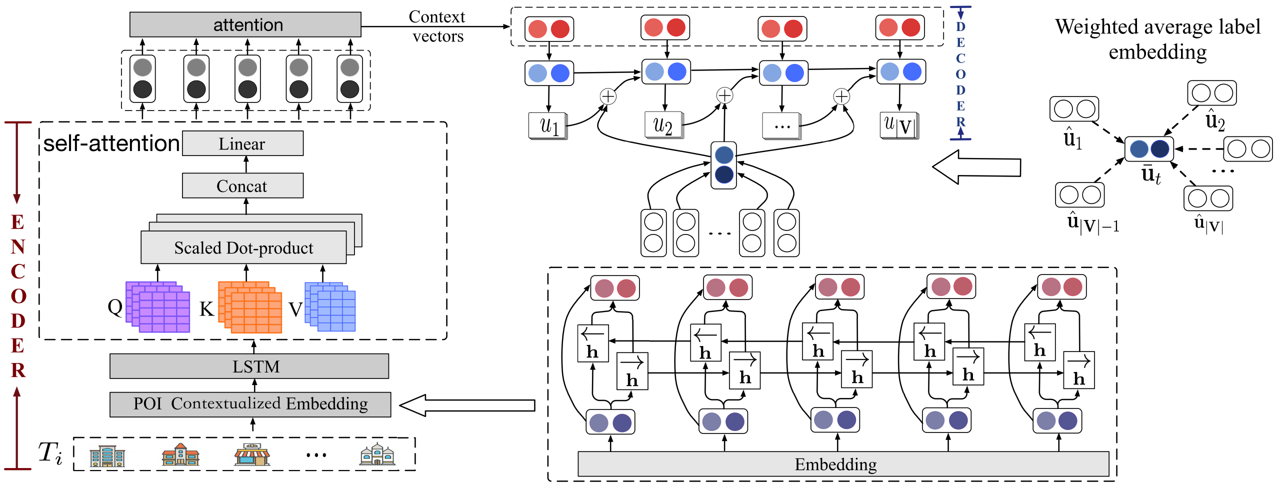


FIGURE 1. Overview of the SCIMAC model.

ordered along the temporal dimension. The trajectories are separated from each other by application dependent thresholds δ_T^j ($1 \leq j \leq m$), which could indicate, for example: – the time span of each T^j is no more than δ_T^j; – the last time-stamp of T^j is at least δ_T^j time units smaller than the time-stamp of the first location in T^{j+1}; etc. For each user, we concatenate all check-in locations to form a single trajectory – subsequently, we divide it into sub-trajectories with the time interval of 6 hours each, as it was done in previous related works [3], [23], [48], [52]. Finally, we add `bos` and `eos` symbols to the front and end of each trajectory respectively.

B. CONTEXT-DEPENDENT POI REPRESENTATION

Inspired by the success of word embedding (e.g., word2vec [24]) in natural language processing, previous location-based models [3], [23], [46] embedded POIs with the surrounding information using CBOW or Skip-Gram models. These, however, only allow a single context-independent representation for each POI and thus cannot distinguish the different contexts information associated with POIs. In words, the POIs learned by the aforementioned works are in accordance with check-in concurrency, but not necessarily conformant with the sequentiallity of the visits. The sequences, though, may reflect different meaning of a same POI in different trajectories. To address this problem, we present a novel contextualized POI embedding model motivated by the ELMo [53] that was originally used for pre-training the language model.

Given a trajectory $T_i = \{c_1, \cdots, c_j, \cdots, c_N\}$ of length N, we use a layered bidirectional LSTMs [26] to model the probability of a check-in c_i given the context in the trajectory. More specifically, we first denote the one-hot representation of i-th check-in c_i as $\mathtt{1}(c_i)$. We then embed c_i into low-dimensional vectors $\mathbf{c}_i$ by multiplying $\mathtt{1}(c_i)$ with an embedding parameter matrix $\mathbf{M} \in \mathbb{R}^{|\mathbf{C}| \times d}$ where d is the dimension of embedding vector and $|\mathbf{C}|$ is size of check-in set $\mathbf{C}$. Subsequently, context-independent vectors $\mathbf{c}_i$ are then passed through L layers of forward LSTMs and L layers backward LSTMs for training.

For each check-in position i, the j-th layer of forward LSTM outputs a context-dependent representation $\overrightarrow{\mathbf{h}}_{i,j}$. A softmax layer is added on the top of the L-th forward LSTM layer to model the probability that predicts the next check-in c_i given the history of T_i:

$$p(c_1, c_2, \ldots, c_N) = \prod_{i=1}^{N} p(c_i | c_1, c_2, \ldots, c_{i-1}) \tag{1}$$

The backward LSTM is very similar to the forward LSTM except that it process check-ins in a reverse way, e.g., the j-th layer of backward LSTM outputs a context-dependent representation $\overleftarrow{\mathbf{h}}_{i,j}$ and a softmax layer based on the L-th layer of backward LSTM predicts c_i given future trajectory $\{c_{i+1}, c_{i+2}, \ldots, c_N\}$:

$$p(c_1, c_2, \ldots, c_N) = \prod_{i=1}^{N} p(c_i | c_{i+1}, c_{i+2}, \ldots, c_N) \tag{2}$$

The bi-directional LSTMs model is trained to maximize the log-likelihood of both the forward and backward directions in an unsupervised way:

$$\begin{aligned}\mathcal{L}_{POI} = \sum_{i=1}^{N} & \log p\left(c_i | c_1, \ldots, c_{i-1}; \Phi_M, \overrightarrow{\Phi}_{\text{LSTM}}, \Phi_s\right) \\ & + \log p\left(c_i | c_{i+1}, \ldots, c_N; \Phi_M, \overleftarrow{\Phi}_{\text{LSTM}}, \Phi_s\right)\end{aligned} \tag{3}$$

where Φ_M, $\overrightarrow{\Phi}_{\text{LSTM}}$ ($\overleftarrow{\Phi}_{\text{LSTM}}$) and Φ_s are the parameters of embedding matrix, forward (backward) LSTMs and softmax layer, respectively.

For each check-in c_i, $2L+1$ representations are computed by bi-directional LSTMs at the pre-training stage:

$$\{\mathbf{c}_i, \overrightarrow{\mathbf{h}}_{i,j}, \overleftarrow{\mathbf{h}}_{i,j} | j \in [1, L]\} = \{\overrightarrow{\mathbf{h}}_{i,j}, \overleftarrow{\mathbf{h}}_{i,j} | j \in [0, L]\} \tag{4}$$

where $\{\overrightarrow{\mathbf{h}}_{i,0} = \overleftarrow{\mathbf{h}}_{i,0} = \mathbf{c}_i\}$ is the embedding layer and $\{\overrightarrow{\mathbf{h}}_{i,j}, \overleftarrow{\mathbf{h}}_{i,j}\}$ are the POI context-dependent representations. Note that one can immediately simplify the model by removing the backward LSTM layers to obtain a lightweight model. However, we empiricially found that the bi-directional embedding described above can obtain better performance on social circle inference.

For simplicity, we use $\mathbf{h}_{i,j}$ to denote $[\overrightarrow{\mathbf{h}}_{i,j} : \overleftarrow{\mathbf{h}}_{i,j}]$, which is the concatenation of forward and backward LSTM representation. We use $\{\mathbf{h}_{i,j}|j = 1, \ldots, L\}$ to denote the embedding vectors of check-in c_i – which will be fed into the next stage of supervised learning, where friends label information is utilized.

C. SOCIAL CIRCLE INFERENCE

We now focus on the social circle inference approach via learning the friend labels.

1) ENCODING TRAJECTORY SEMANTICS

The embedded vectors $\{\mathbf{h}_{i,j}|j = 1, \ldots, L\}$ are trained by L-layers bidirectional LSTMs consisting of the context information regarding the check-in sequence around the i-th position. Instead of leveraging the final layer $\mathbf{h}_{i,L}$ as the representation – in contrast to the previous RNN-based trajectory training models [23], [46], [47] – we utilize a weighted linear combination of *all layers* in learning the context-aware patterns for the purpose of adaptive trajectory representation.

Since we already have embedding vectors $\{\mathbf{h}_{i,j}|j = 1, \ldots, L\}$ for each check-in c_i, we can calculate the weighted average embedding vector at position i (with a note that they are fixed in subsequent supervised learning) as $\hat{\mathbf{z}}_i = \sum_{j=1}^{L} s_j \mathbf{h}_{i,j}$, where s_j are softmax-normalized weights. Obviously, $\hat{\mathbf{z}}_i$ contains information of all $2L$ layers bi-directional LSTMs that can well capture the context-dependent information. This is subsequently concatenated with $\mathbf{c}_i$ – the low dimensional embedding vector of check-in c_i – to obtain the representation $\mathbf{z}_i = [\mathbf{c}_i : \omega_i \hat{\mathbf{z}}_i]$, where ω_i is a scaling factor. Figure 2 depicts the basic idea of learning the context-dependent features from trajectories in an unsupervised manner.

We then utilize a forward LSTM to encode the embedded trajectory and compute sequentially for each check-in with the hidden state $\tilde{\mathbf{h}}_i$:

$$\tilde{\mathbf{h}}_i = \mathrm{LSTM}(\tilde{\mathbf{h}}_{i-1}, \mathbf{z}_i) \tag{5}$$

Due to the diversity of personal behaviors, different users may access particular check-in locations in different orders within the sequence of visits corresponding to their respective trajectories. When processing a particular check-in sequence from a given trajectory, taking the rest of the (relevant) check-in points and their relative distribution to each other into account, will enable capturing implicit information regarding user's motion patterns. This inspires us to incorporate self-attention into trajectory encoder. Self-attention is an attention mechanism relating different positions of a single sequence so

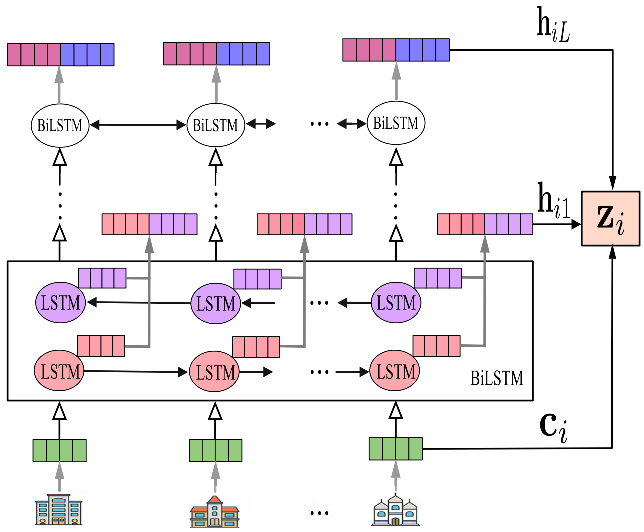


FIGURE 2. Overview of context-aware trajectory learning.

as to compute a representation of the sequence, and has been successfully applied in a variety of tasks including reading comprehension, sentiment classification, machine translation and POI recommendation [50], [54], [55].

The standard self-attention mechanism employs a single attention function with one set query/key/value vector when computing the check-in point corresponding self-attention value. However, we found it beneficial to linearly project the query/key/value vector r times with different and learned linear projections. For each hidden represention of single check-in, we respectively multiply it with three parameter matrices to create the query $\mathbf{q}_i^j$, key $\mathbf{k}_i^j$ and value $\mathbf{v}_i^j$ vectors in each layer of LSTM:

$$\begin{aligned} \mathbf{q}_i^j &= \tilde{\mathbf{h}}_i \mathbf{W}_q^j \\ \mathbf{k}_i^j &= \tilde{\mathbf{h}}_i \mathbf{W}_k^j \\ \mathbf{v}_i^j &= \tilde{\mathbf{h}}_i \mathbf{W}_v^j \end{aligned} \tag{6}$$

where $\mathbf{W}_q^j$, $\mathbf{W}_k^j$, and $\mathbf{W}_v^j$ are parameter matrices in each hidden layer $\tilde{\mathbf{h}}_i$.

Subsequently, we calculate the scores by taking the dot product of query vector of each check-in and key vectors of other related check-ins, which determines how much focus should be placed on other parts of current trajectory when we encode a particular check-in point at a certain position. The next step is to divide the scores by the square root of key vectors' dimension, and then transfer this scalar vector to a softmax function. Now, we obtain the output of current self-attention layer at this position by multiplying the softmax score with value vector:

$$\mathbf{o}_i^j = \mathrm{softmax}\left(\frac{\mathbf{q}_i^j \cdot \mathbf{k}_i^j}{\sqrt{d'}}\right)\mathbf{v}_i^j \tag{7}$$

where d' is the dimension of key vectors. After obtaining $\mathbf{o}_i^j$, which can be computed in parallel, we concatenate the output vectors and multiply them by an additional weights matrix $\mathbf{W}$.

Here, we denote the final representation of each check-in passing multi-layers self-attention as $\mathbf{f}_i$, which can be calculated by:

$$\mathbf{f}_i = [\mathbf{o}_i^1, \ldots, \mathbf{o}_i^r]\mathbf{W} \tag{8}$$

Intuitively, not all check-ins make the same contribution in predicting friend labels. In natural language processing, the attention mechanism [56] computes context vector focusing on a set of positions where the most relevant information is concentrated. Similarly, we compute the context vector $\mathbf{a}_t$ which focuses on a set of most relevant check-ins at time-step t as follows:

$$\begin{aligned} u_{ti} &= \mathbf{W}_3^{\mathsf{T}} \tanh(\mathbf{W}_1\mathbf{f}_i + \mathbf{W}_2\mathbf{s}_t) \\ \alpha_{ti} &= \mathrm{softmax}(u_{ti}) \\ \mathbf{a}_t &= \sum_i \alpha_{ti}\mathbf{f}_i \end{aligned} \tag{9}$$

where $\mathbf{W}_1$, $\mathbf{W}_2$ and $\mathbf{W}_3$ are weight parameters to be learned, $\mathbf{f}_i$ is the representation of self-attention and $\mathbf{s}_t$ is the hidden states of the decoder at time-step t. Weight α_{ti} is the i-th hidden state of the encoder used for computing the context vector $\mathbf{a}_t$, which is then passed to the decoder at time-step t.

2) CIRCLE INFERENCE

The decoder uses another LSTM to compute the hidden states $\mathbf{s}_t$ and utilizes softmax to infer the users' probability distribution over the label space. Rather than treating each label independently in TSCI [23], SCIMAC explicitly models the label correlations and performs the circle inference in a unified manner.

For all users (label space) $\mathbf{V} = (u_1, u_2, \ldots, u_{|\mathbf{V}|})$, we first sort them according to their frequency in a descending order and add `bol` and `eol` symbols at the beginning and the end of the sequence respectively. We embed all the user labels in a low dimensional space by an embedding matrix $\mathbf{M} \in \mathbb{R}^{|\mathbf{V}| \times d''}$, where d'' is the dimension of the embedded vectors. We denote the probability distribution vector over the label space at time-step t as $\mathbf{u}_t$ and it is computed as follows:

$$\mathbf{s}_t = \mathrm{LSTM}\,(\mathbf{s}_{t-1}, [\hat{\mathbf{u}}_{t-1}; \mathbf{a}_{t-1}]) \tag{10}$$

$$\mathbf{g}_t = \mathbf{W}_g \tanh(\mathbf{W}_4\mathbf{a}_t + \mathbf{W}_5\mathbf{s}_t) \tag{11}$$

$$\mathbf{u}_t = \mathrm{softmax}(\mathbf{g}_t + \mathbf{M}_t) \tag{12}$$

where $\mathbf{s}_t$ is the hidden state; $\mathbf{W}_g$, $\mathbf{W}_4$ and $\mathbf{W}_5$ are weight parameters; $\hat{\mathbf{u}}_{t-1}$ is the embedding of the user that has highest probability under the distribution $\mathbf{u}_{t-1}$ predicted at time-step $t-1$; $[\hat{\mathbf{u}}_{t-1}; \mathbf{a}_{t-1}]$ is the concatenation of $\hat{\mathbf{u}}_{t-1}$ and attention vector $\mathbf{a}_{t-1}$ at previous time step; and $\mathbf{M}_t$ is the mask vector at time-step t and is used to prevent predicting repeated labels in $\mathbf{u}_t$.

During training, the i-th element in vector $\mathbf{u}_t$ is the predicted probability that user u_i is the friend for current trajectory at time-step t. In addition, we compute the mask vector $\mathbf{M}_t$ in this way: $\mathbf{M}_1$ is initialized as all zero vector and then at time-step t the i-th element of $\mathbf{M}_t$ is set to ∞ if label u_i has been predicted in the previous t-1 steps. This label sequence embedding has been successfully used in text classification [57]. However, we note that the SCIM problem is more complex than multi-label text classification, where the number of classes is usually small. In contrast, we encounter large number of labels in SCIM, i.e., the label space for each trajectory is the entire label set.

Finally, the objective of training is to minimize the categorical cross-entropy loss on the labeled trajectories:

$$\mathcal{L}(\theta) = -\frac{1}{M}\sum_{i=1}^{M}\sum_{j=1}^{|\mathbf{V}|} \mathbb{1}(u_j \in \mathbf{U}_i) \log p(u_j \in \mathbf{U}_i) \tag{13}$$

where indicator function $\mathbb{1}(\cdot)$ means the j-th user is the friend of trajectory T_i – it equals 1 if u_j is one of the labels of T_i and equals 0 otherwise, and $p(\cdot)$ is the probability based on $\mathbf{u}_t$ that predicted label u_j belongs to T_i.

At the inference stage, beam search is utilized to find top-ranked prediction path. In many seq2seq encoder-decoder models such as neural machine translation [56], beam search is the de facto method employed for decoder at inference time. Beam search maintains B top-ranked prediction beams which are a sequence of user labels that are predicted at $t-1$ time step. At time step t, every possible user label, except those already in the beams, is added to each existing beam. We select B top-ranked beams from these expanded beams according to probability distribution of the trained model. We repeat the procedure until an `eol` symbol is predicted or the beam length reaches the preset maximum length for all the B top-ranked beams, and then move the beam to the candidate beam set. Finally, we choose the one with highest probability from the candidate beam set that contains top-ranked beam ending with `eol`. Algorithm 1 illustrates the training of SCIMAC.

3) OVERCOMING THE INFERENCE BIAS

In Eq.(10), $\hat{\mathbf{u}}_{t-1}$ is the embedding of the user label with the highest probability according to the probability distribution vector $\mathbf{u}_{t-1}$ but other less likely users are not considered. If there is a wrong prediction at time-step t, it would result in wrong predictions at the following time steps. This phenomeno is also known as *exposure bias* problem in multi-label text classification [57]. Obviously, the beam search mechanism mentioned above could, to certain extent, alleviate the bias by selecting from a number of prediction paths at the inference stage. However, it cannot fundamentally solve the problem due to the choice of highest probability label at each time step. As a result, the bias prediction could occur for every possible inference path. On the other hand, this problem could be largely mitigated if we take all elements in probability distribution vector $\mathbf{u}_{t-1}$ into account at the training stage [57], as illustrated in Figure 3. Let $\bar{\mathbf{u}}_{t-1}$ denote the weighted average label embedding computed as follows:

$$\bar{\mathbf{u}}_{t-1} = \sum_{i=1}^{|\mathbf{V}|} u_{t-1}^{(i)} \hat{\mathbf{u}}_i \tag{14}$$

Algorithm 1 Training of SCIMAC

Input: Trajectory: $T_i \in \Gamma$; User set $\mathbf{U}_i \subseteq \mathbf{V}$.

/* Contextualized Embedding */

foreach *check-in* $c_j \in T_i$ **do**
  Get the one-hot representation $1(c_j)$;
  Multiply a embedding matrix to compute the low dimensional embedding $\mathbf{c}_i$;
  Feed the vector $\mathbf{c}_i$ to L layers Bi-LSTMs to obtain $\{\mathbf{h}_{i,j} | j = 1, \ldots, L\}$;
  Maximize $\mathcal{L}_{POI}$(Eq.(3)).
end

/* Initial User Label Embedding */

foreach *user label* $u_i \in \mathbf{V}$ **do**
  Multiply the embedding matrix to get corresponding vector $\hat{\mathbf{u}}_i$.
end

/* Training */

Training dataset $\mathcal{D} \leftarrow \emptyset$.
foreach $\mathbf{U}_i \subseteq \mathbf{V}$ **do**
  $\mathcal{D} \leftarrow < T_i, \mathbf{U}_i >$.
end
repeat
  foreach $< T_i, \mathbf{U}_i > \in \mathcal{D}$ **do**
    foreach $c_j \in T_i$ **do**
      Calculate the weighted average embedding vector $\mathbf{z}_i$ by concatenating $\mathbf{c}_i$ and $\hat{\mathbf{z}}_i$;
      Encode the embeded check-in $\mathbf{z}_i$ to obtain hidden state $\tilde{\mathbf{h}}_i$;
      Employ self-attention to learn various aspects of users' preference via r sets of query/key/value vector computing the multi-layers representation $\mathbf{f}_i$.
    end
    Compute the context vector $\mathbf{a}_t$ which focuses on a set of most relevant check-ins at time step t.
    Predict label u_i via LSTM(Eq.(10)-(12)).
    Minimize $\mathcal{L}(\theta)$ (Eq.(13)).
  end
until *converge*

Output: Training Model $\mathcal{M}$.

where $u_{t-1}^{(i)}$ is the i-th element of $\mathbf{u}_{t-1}$ and $\hat{\mathbf{u}}_i$ is the embedding vector of i-th user label. Obviously, $\bar{\mathbf{u}}_t$ contains information of all possible labels and probability distribution of these labels at time-step $t-1$. Therefore, instead of only passing $\hat{\mathbf{u}}_{t-1}$ to decoder, we reformulate Eq.(10) by considering all informative labels:

$$\mathbf{s}_t = \text{LSTM}(\mathbf{s}_{t-1}, [(1-\lambda)\hat{\mathbf{u}}_{t-1} + \lambda\bar{\mathbf{u}}_{t-1}; \mathbf{a}_{t-1}]) \tag{15}$$

where λ is a hyper-parameter controlling the compromise between weighted average label embedding and individual label embedding. The hidden state $\mathbf{s}_t$ is computed based on information of all possible labels. By considering every label and their possibilities, the impact of single mis-prediction can be attenuated in subsequent friend inference. The procedure for contextualized POI embedding is listed in Algorithm 2.

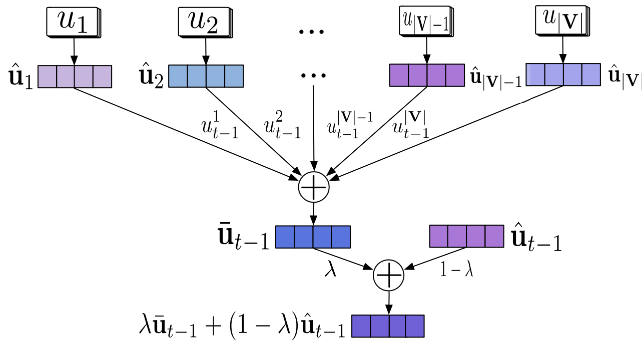


FIGURE 3. Illustration of label embedding.

Algorithm 2 Label Embedding

Input: Predicted probability distribution vector $\mathbf{u}_{t-1}$ at time step $t-1$; embedding vector $\hat{\mathbf{u}}_i$ of each user label.

Initialize the average label embedding vector $\bar{\mathbf{u}}_{t-1}$ with 0.
foreach $u_{t-1}^{(i)} \in \mathbf{u}_{t-1}$ **do**
  Update $\bar{\mathbf{u}}_{t-1} \leftarrow \bar{\mathbf{u}}_{t-1} + u_{t-1}^{(i)}\hat{\mathbf{u}}_i$.
end
Find the embedding $\hat{\mathbf{u}}_{t-1}$ of label with the highest probability according to $\mathbf{u}_{t-1}$.
Balance $\bar{\mathbf{u}}_t$ and $\hat{\mathbf{u}}_{t-1}$ in Eq.(15) with λ.

Output: Label embedding $(1-\lambda)\hat{\mathbf{u}}_{t-1} + \lambda\bar{\mathbf{u}}_{t-1}$.

V. EVALUATION

We now present the details of the evaluation of the performance of our SCIMAC model in comparison with the state-of-the-art social circle inference methods on four real-world datasets. Specifically, we focus on the following three main quantitative observations:

- *QO1.* What is the *effectiveness* of SCIMAC – i.e., does it provide better circle inference performance compared to the existing approaches/baselines?
- *QO2.* How important is the impact of the three components – context-aware embedding, self-attention and label embedding – in our SCIMAC model?
- *QO3.* How do the parameter settings affect the performance of our model?

A. DATASETS

We conduct experiments on four publicly available datasets: Brightkite, Gowalla [58], Tokyo@Foursquare and New York@Foursquare, where the two most popular cities Tokyo and New York are extracted from Foursquare dataset [38], [59]. All datasets consist of user check-in history and social relations.

We evaluate the methods on both of the SCIM tasks: SCIM-I and SCIM-II (cf.Section III): (1) we chose 50% of sub-trajectories of each user for training, and the remainder for testing for SCIM-I task; and (2) for SCIM-II task, we infer the social circle members for anonymous trajectories which have not appeared in the training set. The data pre-processing is exactly the same in [23]:

- For *Brightkite and Gowalla* datasets, we linked the label set (circle members) to each sub-trajectory and randomly selected 201 and 92 users for SCIM-I, respectively. In the SCIM-II task, we selected the users having at least 5 friends and obtained 199 users for Brightkite and 514 users for Gowalla.
- For *Tokyo and New York @ Foursquare*, we constructed the social relations using the method proposed in [38]: we randomly chose 60 seeding users for Tokyo and 40 for New York, which therefore becomes the seed-user set. We then explored the social network to find users who have at least 5 friends in the user set. Among these users, we then randomly chose 200 and 150, as well as their motion traces, as the final data for the two datasets. The trajectories generated by users who have at least 5 friends but not used in SCIM-I task would be used in SCIM-II task.

The statistics of datasets after pre-processing are shown in Table 2.

TABLE 2. Dataset statistics. |V|: The number of users; U_{I}: The number of users for SCIM-I task; U_{II}: The number of users for SCIM-II task; T_{train}/T_{test}: The number of trajectories in training vs. the number of trajectories in testing for SCIM-I task; T_{II}: The number of sub-trajectories for SCIM-II task.

Dataset	$\lvert\mathbf{V}\rvert$	U_{I}	U_{II}	$T_{\text{train}}/T_{\text{test}}$	T_{II}
Brightkite	92	92	199	10,012/10,061	37,403
Gowalla	201	201	514	10,104/10,052	59,514
Tokyo	60	200	248	9,036/9,153	22,769
New York	40	150	62	6,220/6,295	5,185

B. METRICS

Following the existing related works [23], [60], [61], we selected three standard metrics to comparatively evaluate the multi-label classification performance of our model and baselines: *macro-Recall (macro-R), macro-F1 and Accuracy*. While macro-R is the average proportion of predicted circle members that are also in the ground truth, macro-F1 is defined as the harmonic mean of macro-Precision (macro-P) and macro-R. Accuracy reflects the correctly predicted friends. Specifically, they are formally defined as:

$$\text{macro-P} = \frac{1}{Q}\sum \frac{\#\text{ correctly predicted friends}}{\#\text{ predicted friends}}$$
$$\text{macro-R} = \frac{1}{Q}\sum \frac{\#\text{ correctly predicted friends}}{\#\text{ True friends}}$$
$$\text{macro-F1} = \frac{2 \times (\text{macro-P}) \times (\text{macro-R})}{(\text{macro-P}) + (\text{macro-R})}$$
$$\text{Accuracy} = \frac{1}{Q}\sum \frac{\#\text{ correctly predicted friends}}{\#\text{ True Friends} \cup \#\text{ predicted friends}}$$

where Q denotes the number of trajectories in the testing set T_{test}.

C. BASELINES

The comparison of the advantages of our model was done with respect to the following baselines:

- *SVM for multi-label classification*: In accordance with [23], [62], we trained a linear kernel based SVM model for circle inference.
- *Matrix Factorization (MF)*: We construct a friend check-in frequency matrix where each cell represents the number of times the corresponding check-in has been visited by that user in all trajectories. To obtain the friend list for a new testing trajectory, we calculate the similarity between its vector and every friend vector which are obtained from the matrix factorization [37].
- *Co-visit* [38]: For training data, we concatenate the trajectories which have the same friendship label set, assuring that each user associates with a trajectory. Then for a new testing trajectory, we identify its friends from similar trajectories in terms of common check-ins in the training data, where Longest Common Subsequence (LCS) technique is used to find user common interest or locations. The threshold for the number of co-visits is manually optimized.
- *MLP*: To show the performance of deep learning based approach in terms of capturing the spatio-temporal information and multi-label classification, a multi-layer perceptron (MLP) based model proposed in [52] is trained in terms of social circle inference.
- *DeepMIML* [52]: DeepMIML is a multi-instance multi-label classification method originally developed for language and image classification. Since the corresponding codes have not been published yet, we re-implement it using Auto-encoder to obtain the representation vector of trajectories and a 2D sub-concept layer proposed in DeepMIML to learn social circles for a given trajectory.
- *TULER* [3]: One of the most successful model on identifying human mobility patterns proposed in [3], which leverages RNNs to capture the sequential patterns of human trajectories and to predict the generators of unknown trajectories. We train the TULER using a stacked GRU for the multi-label classification problem.
- *walk2friends* [22] is a word2vec-based model that employs mobility information for inferring social links, relying on neural networks to learn the location and trajectory representation. It utilizes pairwise similarity measures to compare two users' mobility patterns and judge whether they are socially related or not.
- *O2O-Inf* [20] is to infer online social ties using offline geographical activities of users by feature modeling and link prediction. Feature modeling is to characterize both direct and indirect geographical interactions between nodes from co-location and graph features, while link prediction is performed to infer the social ties based on the observed social links.
- *DeepTSCI* [23] is a SCIM model using human mobility patterns for inferring corresponding social circles. While the original DeepTSCI used bi-LSTM, Autoencoder and

TABLE 3. Performance comparison among different algorithms for SCIM-I on four datasets.

Method	Brightkite			Gowalla			Tokyo			New York		
	Macro-R	Macro-F1	Accuracy	Macro-R	Macro-F1	Accuracy	Macro-R	Macro-F1	Accuracy	Macro-R	Macr.o-F1	Accuracy
SVM	0.6348	0.6870	0.5404	0.4815	0.6218	0.4467	0.3442	0.4601	0.3246	0.3652	0.4893	0.3463
MF	0.5253	0.4121	0.2681	0.4600	0.2075	0.1122	0.2997	0.3119	0.1908	0.4608	0.4040	0.2774
Co-Visit	0.5542	0.4207	0.3210	0.4144	0.3710	0.2941	0.4592	0.4463	0.3790	0.6209	0.5902	0.5339
MLP	0.6524	0.6954	0.5547	0.5126	0.6288	0.4505	0.4766	0.5617	0.4401	0.4887	0.5903	0.4529
DeepMIML	0.5362	0.6371	0.4650	0.5239	0.5780	0.4022	0.5633	0.6010	0.5111	0.6480	0.7156	0.6273
walk2friends	0.6471	0.6635	0.5287	0.4935	0.5671	0.4412	0.5160	0.5432	0.4879	0.5346	0.5781	0.5034
O2O-Inf	0.5681	0.4312	0.3541	0.4322	0.4011	0.3233	0.4887	0.4715	0.4002	0.5162	0.5343	0.4987
TULER	0.6203	0.6896	0.5350	0.4697	0.6176	0.4358	0.5592	0.6282	0.5318	0.6487	0.7174	0.6300
DeepTSCI	0.6999	0.7320	0.6061	0.6139	0.6851	0.5409	0.6022	0.6534	0.5704	0.6757	0.7220	0.6407
SCIMAC	**0.8149**	**0.8095**	**0.7311**	**0.6758**	**0.6909**	**0.5709**	**0.7020**	**0.7027**	**0.6596**	**0.8277**	**0.8093**	**0.7596**

TABLE 4. Performance comparison among different algorithms for SCIM-II on four datasets.

Method	Brightkite			Gowalla			Tokyo			New York		
	Macro-R	Macro-F1	Accuracy	Macro-R	Macro-F1	Accuracy	Macro-R	Macro-F1	Accuracy	Macro-R	Macro-F1	Accuracy
SVM	0.1736	0.1615	0.1032	0.1253	0.1878	0.0970	0.0901	0.1260	0.0643	0.1591	0.2271	0.1273
MF	0.1846	0.1006	0.0641	0.1114	0.0959	0.0511	0.1154	0.1130	0.0653	0.1051	0.0914	0.0558
Co-Visit	0.1452	0.0542	0.0293	0.1010	0.0870	0.0420	0.1008	0.0894	0.0491	0.1172	0.0905	0.0522
MLP	0.1584	0.1709	0.0855	0.1519	0.2101	0.1045	0.1183	0.1477	0.0778	0.2143	0.2707	0.1542
DeepMIML	0.2847	0.2482	0.1240	0.1524	0.1974	0.0989	0.1251	0.1579	0.0820	0.2433	0.2909	0.1620
walk2friends	0.2112	0.1734	0.1153	0.1498	0.1871	0.0973	0.1241	0.1489	0.0799	0.2241	0.2673	0.1583
O2O-Inf	0.1341	0.0862	0.0547	0.1211	0.0884	0.0520	0.0935	0.0776	0.0473	0.1011	0.1064	0.0759
TULER	0.3348	0.2525	0.1371	0.1377	0.2009	0.0999	0.1233	0.1554	0.0808	0.2489	0.2943	0.1661
DeepTSCI	0.3541	0.2608	0.1404	0.1743	**0.2271**	0.1141	0.1386	0.1676	0.0902	0.2760	0.3101	0.1784
SCIMAC	**0.3587**	**0.2688**	**0.1543**	**0.2057**	0.2154	**0.1154**	**0.2195**	**0.2206**	**0.1280**	**0.3383**	**0.3117**	**0.1900**

VAE to learn the latent representations of trajectories, respectively, we choose the the best performed one (based on VAE) for comparisons in our experiments.

The deep learning based models, including DeepTSCI, TULER, MLP, Co-visit, walk2friends, O2O-Inf, Deep MIML and our SCIMAC, were implemented on PyTorch with a GTX1080Ti GPU, while the traditional methods (SVM and MF) were implemented with scikit-learn library.

Parameter Settings: The learning rate of all models is initialized with 0.001 and decays with a rate of 0.9. The activation function for all methods is ReLU, and the dropout rate is set to 0.5 while the batch size is 16 for all RNN based models following [23]. We embed each POI into a 512 dimensional vector, and use two one-hot vectors for representing the temporal and spatial information. Furthermore, we use 300 neuron units for the classifier, and 256 units for encoders and 512 units decoders for DeepTSCI and SCIMAC. The hyper-parameter λ in SCIMAC is empirically tuned to 0.5 in all experiments.

D. OVERALL PERFORMANCE (QO 1)

Table 3 and Table 4 show the performance comparison among our model and baselines for the two SCIM tasks in terms of the three metrics defined (cf. Section V-B), where the best performance is shown in **boldface**, while the second best in underlined font.

1) ON SCIM-I

We can observe in Table 3 that the proposed SCIMAC significantly outperforms the other 9 approaches across all datasets. For example, compared to DeepTSCI – the best approach in baselines, it achieves 16.43%, 10.58% and 20.62% higher (on average) over the best DeepTSCI in terms of macro-R, macro-F1 and accuracy on Brightkite dataset. The reason SCIMAC achieved significant improvements on circle prediction is three-fold. First, it uses a contextualized check-in representation model which distinguishes the POI embedding in different trajectories. Second, a self-attention module is utilized in SCIMAC to learn different aspects of users' preference over POIs. Third, SCIMAC explicitly models the latent correlations among the users when inferring the circles.

Among the baselines, DeepTSCI shows the best performance which demonstrates the effectiveness of learning stochastic latent factors with variational autoencoders. Two recent works walk2friends and O2O-inf did not show competitive results, which is not surprise because: (1) walk2friends is a model using representation learning for measuring the mobility similarity, which, however, uses context-independent embedding and fails to learn the contextualized semantics in different contexts; and (2) O2O-inf, in contrast, is a traditional feature-based method requiring hand-craft feature engineering and partial social interactions for learning human mobility, and therefore cannot exhibit competitive performance without social information in our implements.

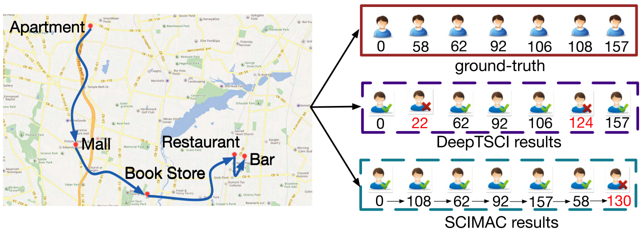


FIGURE 4. Visualization of a SCIM-II result on New York dataset. Left: The trajectory needs to be classified. Upper right: The ground-truth circle; Middle right: The results of DeepTSCI; Bottom right: The results of SCIMAC.

2) ON SCIM-II

As illustrated in Table 4, SCIMAC also consistently exhibits superior performance on SCIM-II task over the baselines across four datasets. Obviously, SCIM-II task is more difficult than SCIM-I, since the owners of the testing trajectories are anonymous – i.e., there is no training data for these users. Figure 4 plots a SCIM-II result on New York dataset using DeepTSCI and SCIMAC, where we used Baidu Map[2] for locating the POIs. We can see that SCIMAC infers the circle member sequentially while DeepTSCI predicts the labels independently. Apparently, by modeling the implicit correlations between labels, SCIMAC outperforms DeepTSCI on social circle inference. The improvement is straightforward: while the test trajectory is a cold-start data sample, the social circle members are not, which means their mobility patterns have been learned by the model and their correlations can be implicitly constructed through learning the motion pattern similarity.

Essentially, the task of SCIM-II is to measure the capability of learning mobility patterns and to infer the circles based only on the motion similarity among users – the motivation is that two online friends should share similar offline geographical activities. Although it can be used to address the cold-start problem for SCIM, we note that there is a bottleneck in SCIM-II task: the online users whole are in the same community or social circle may not have similar motion patterns, which has also been observed in spatio-temporal mining works [58]. An immediate solution to improve the performance on SCIM-II is to impose more spatio-temporal constraints on possible circle inference, e.g., two friends may have similar geographical motions if they are living in a same city, which, however, is beyond the scope of this work and is left for our future work.

3) EFFICIENCY

Efficiency is another important criteria in social circle inference. Here we compare the efficiency of SCIMAC against

[2]https://map.baidu.com

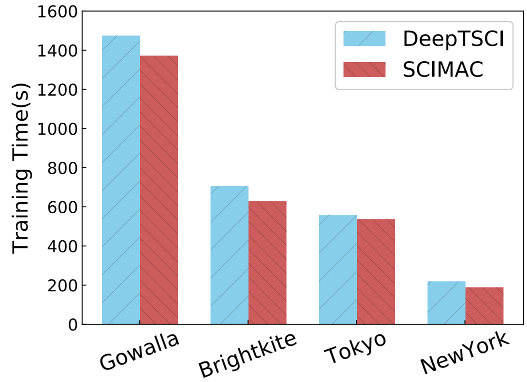


FIGURE 5. Efficiency comparison between SCIMAC and DeepTSCI.

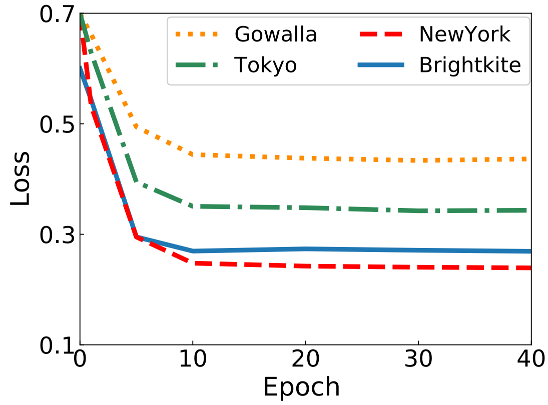


FIGURE 6. Training procedure of SCIMAC on SCIM-I task. The trend on SCIM-II task is similar and is omitted for simplicity.

DeepTSCI empirically. Both of the two models require pre-training, trajectory learning and multi-label classification. However, they are quite different in each part. During pre-training, DeepTSCI leverages VAE, combined with bi-directional LSTM to learn probabilistic latent patterns in trajectories, which requires significant more time for training. In contrast, SCIMAC uses bi-directional LSTM to learn the context-dependent features and hierarchical motion patterns for POI representation, which is more efficient compared to stochastic inference in DeepTSCI. When learning users' mobility patterns, DeepTSCI employs a self-attention layer for intra-motion pattern learning requiring slightly more time which is negligible by utilizing the parallel computation of GPU. Finally, DeepTSCI explicitly models correlations among labels when inferring the circles, which requires additional computation on label embedding and sequential inference. As illustrated in Figure 5, the computational cost for two models remains in the same level, while SCIMAC is slightly efficient than DeepTSCI. Note that we omit the comparison to other models since they are either too simple on learning trajectory patterns (e.g., walk2friend and MLP), or require considerable too much time on training (e.g., SVM and MF) due to without the acceleration with GPUs.

4) MODEL TRAINING

Figure 6 plots the training procedure of SCIMAC on four datasets, from which we can clearly see that our model

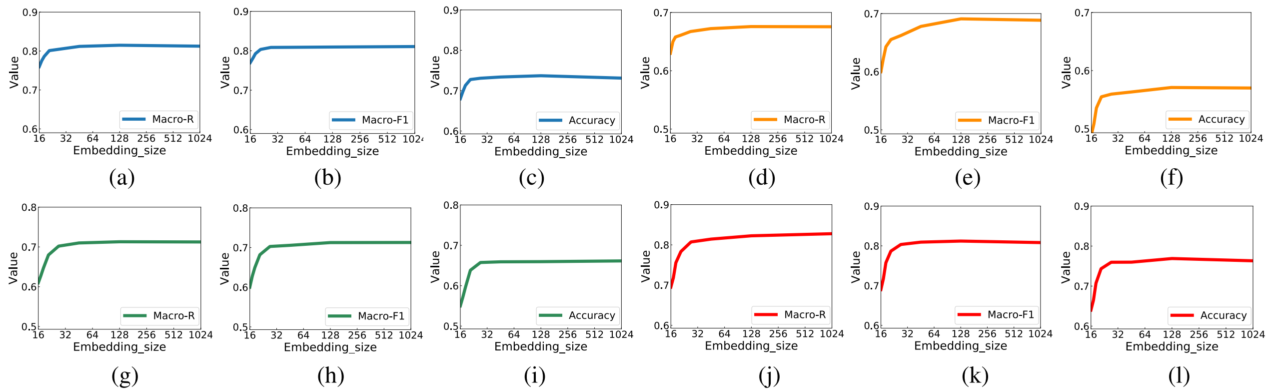


FIGURE 7. Impact of parameters on SCIM-I task. The impact on SCIM-II task is similar and is omitted for simplicity. (a) Brightkite (R). (b) Brightkite (F1). (c) Brightkite (Acc). (d) Gowalla (R). (e) Gowalla (F1). (f) Gowalla (Acc). (g) Tokyo (R). (h) Tokyo (F1). (i) Tokyo (Acc). (j) NY (R). (k) NY (F1). (l) NY (Acc).

converge very fast, e.g., usually 10 epochs are enough for it to achieve the best performance. The main reason behind this phenomena is that pre-training stage can learn a good POI representation in a context-aware manner and has captured the transition patterns in trajectories and adequately reflects the long-short dependencies among check-ins. Consequently, the trajectory learning module (LSTM) requires less time to train the model. In addition, the self-attention layer also helps reducing the training time mainly due to that the various aspects of user's preference over POIs have been combined for training the model.

5) PARAMETER SENSITIVITY (QO 3)

We optimized our SCIMAC model by varying some important parameters, e.g., the sizes of the POI embedding, the number of LSTM layers and batch size, etc. When tuning these parameters, we did not observe too much improvement by stacking deeper layers and a 2-layer LSTM is sufficient for SCIMAC to model the user sequential check-ins. We also empirically found that 512-dimension is sufficient for our model and it may suffer from slight over-fitting problem (c.f Figure 7) when the embedding size beyond 512, e.g., on New York dataset.

E. ABLATION STUDY (QO 2)

To study the utility of each component in the SCIMAC model, we decomposed our model with a set of variants, including the following:

- *Without contextualized embedding (SCIMA):* It prunes the context-dependent pretraining for POI representation and substitutes with a word2vec embedding for POIs, as in previous works [22], [23].
- *The impact of label embedding and self-attention:* To investigate the effect of the two components, we design an experiment that showing their impact respectively.

The utility of contextualized embedding are illustrated in Figure 8. As we can see, SCIMAC, the model with contextualized embedding, performs better than the pruned

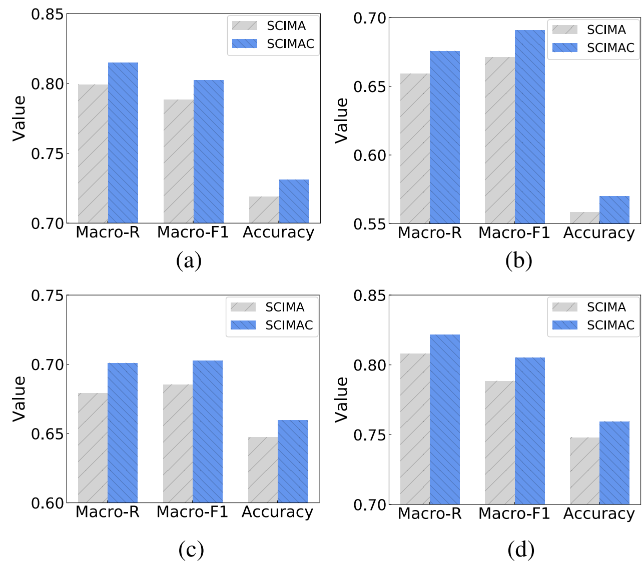


FIGURE 8. Effect of contextualized embedding in SCIMAC. (a) Results on Brightkite. (b) Results on Gowalla. (c) Results on Tokyo. (d) Results on New York.

version SCIMA that utilizing context-independent representation, which demonstrates the effect of learning context-aware embedding of POIs in the pre-training stage. Recall that SCIMAC used stacked bi-directional LSTM for learning hierarchical representation for POIs with the similar structure of ELMo [53], where the syntactic information is better represented at lower layers while semantic information is captured by higher layers. In contrast, there is no such explicit difference in POI representation and mobility learning, although the same check-ins have different meanings for different people and the contextualized embedding indeed helps learning better representations.

Figure 9 illustrates the impact of label embedding and self-attention on the performance of SCIMAC, where we used contextualized embedding in this experiment. Obviously, both of the two components play important but different roles in inferring the circle. Label embedding is to infer the members sequentially by considering the implicit correlations

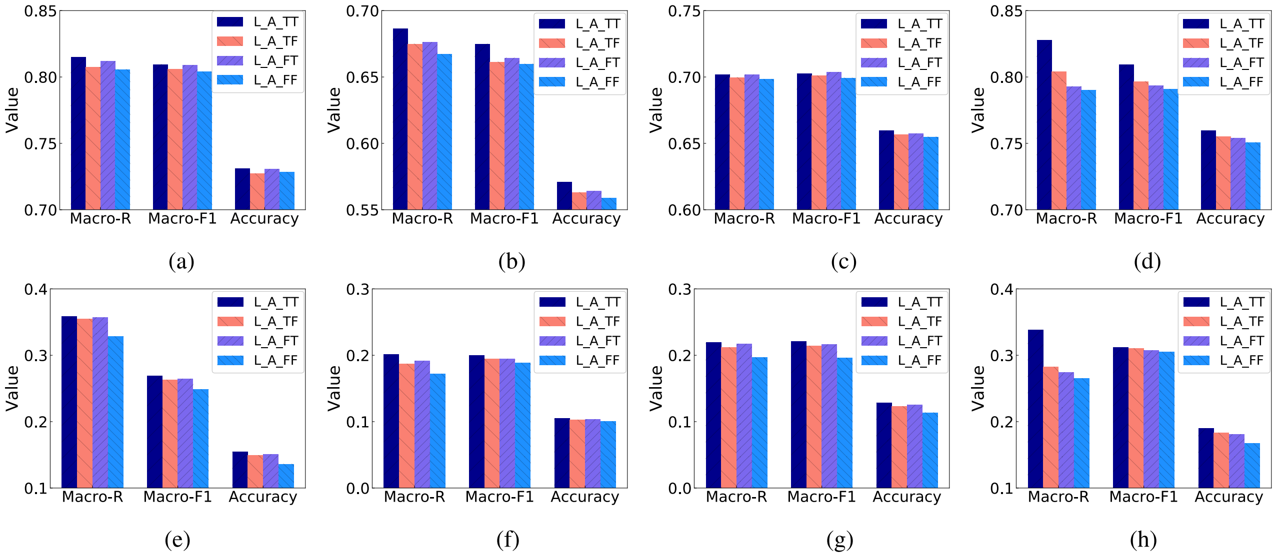


FIGURE 9. Impact of label embedding and self-attention. Letters "TF" in "L_A_TF" are True and False meaning whether using Label embedding and self-Attention in SCIMAC, respectively. (a) SCIM-I on Brightkite. (b) SCIM-I on Gowalla. (c) SCIM-I on Tokyo. (d) SCIM-I on New York. (e) SCIM-II on Brightkite. (f) SCIM-II on Gowalla. (g) SCIM-II on Tokyo. (h) SCIM-II on New York.

among the members, while self-attention allows the model to extract different aspects of a motion trace into multiple vector-representations. The gap between "L_A_TT" and "L_A_FT" shows the performance gain of label embedding, which is not apparent for datasets with larger value of nodes; however, it becomes significant for datasets with smaller nodes, e.g., New York dataset. This is reasonable due to the sequential inference in SCIMAC, which, as we discussed in Section IV-C.3, may result in bias problem if there is a wrong prediction. This phenomena becomes more severe when there are more circle members need to be inferred, and therefore to certain extent compromises the performance gain of modeling implicit correlations among users with label embedding. On the other hand, we can observe the effect of self-attention by investigating the gap between "L_A_TT" and "L_A_TF", which validates our motivation of focusing on the most relevant check-ins in a trajectory and capturing semantic features with (self-)attention mechanism.

VI. CONCLUDING REMARKS

We presented a new social circle inference framework, which is built upon the readily available user mobility data, without requiring any explicit online network kind of information. Comparing with the ubiquitous social circle inference methods that require users' online interaction data, our method has the significant advantage of inferring the user relationships from auxiliary information, which can also complement the previous approaches – i.e., in order to improve the inference performance in certain settings in which the network information is available. Using pre-trained trajectory models to combine the hierarchical layered semantics of RNN to represent context-dependent POIs, our model is capable of incorporating multiple meanings of user check-ins. Our proposed framework seamlessly integrates various aspects of users' preference over POIs with a self-attention module, which significantly improves the effectiveness on learning human mobility patterns by distinguishing the importance of check-ins in both intra- and inter-trajectory learning. With embedding the labels in a low-dimensional space, we construct our model on the implicit user space to alleviate the problem of inference bias when classifying the trajectories. Extensive experiments have been conducted to show the superior performance of our model compared with previous SCIM methods.

One of our immediate future work is to learn other user contents for inferring the social circles, such as profile and published data (e.g., text and pictures). In addition, an open question is how to defend against the inference attack raised by SCIM. Common solutions, such as random perturbation or filtering out important check-in observations, may degrade the performance of beneficial machine learning tasks. As of our future work, we plan to investigate the novel defense mechanisms against the inference attack by leveraging the deep generative models to generate learning performance guaranteed synthesis mobility data while preserving the anonymization of online social relations.

VII. ACKNOWLEDGMENTS

This work was supported by National Natural Science Foundation of China (Grant No.61602097 and No.61472064), NSF grants III 1213038 and CNS 1646107, and ONR grant N00014-14-10215.

REFERENCES

[1] J. Manotumruksa, C. Macdonald, and I. Ounis, "A contextual attention recurrent architecture for context-aware venue recommendation," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jun. 2018. pp. 555–564.

[2] D. Chen, C. S. Ong, and L. Xie, "Learning points and routes to recommend trajectories," in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2016, pp. 2227–2232.

[3] Q. Gao, F. Zhou, K. Zhang, G. Trajcevski, X. Luo, and F. Zhang, "Identifying human mobility via trajectory embeddings," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 1689–1695.
[4] F. Zhou, Q. Gao, G. Trajcevski, K. Zhang, T. Zhong, and F. Zhang, "Trajectory-user linking via variational autoencoder," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 3212–3218.
[5] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, Nov. 2003, pp. 556–559.
[6] C. Yang, M. Sun, W. X. Zhao, Z. Liu, and E. Y. Chang, "A neural network approach to jointly modeling social networks and mobile trajectories," *ACM Trans. Inf. Syst.*, vol. 35, no. 4, p. 36, Aug. 2017.
[7] D. Ding, M. Zhang, S.-Y. Li, J. Tang, X. Chen, and Z.-H. Zhou, "Baydnn: Friend recommendation with Bayesian personalized ranking deep neural network," in *Proc. ACM Conf. Inf. Knowl. Manage.*, Nov. 2017, pp. 1479–1488.
[8] T. Ebesu and Y. Fang, "Neural citation network for context-aware citation recommendation," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Aug. 2017, pp. 1093–1096.
[9] Y. Ruan, D. Fuhry, and S. Parthasarathy, "Efficient community detection in large networks using content and links," in *Proc. 22nd Int. Conf. World Wide Web*, May 2013, pp. 1089–1098.
[10] E. Shaabani, A. Aleali, P. Shakarian, and J. Bertetto, "Early identification of violent criminal gang members," in *Proc. 21st ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2015, pp. 2079–2088.
[11] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2014, pp. 701–710.
[12] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *Proc. 24th Int. Conf. World Wide Web*, May 2015, pp. 1067–1077.
[13] L. F. Ribeiro, P. H. Saverese, and D. R. Figueiredo, "Struc2vec: Learning node representations from structural identity," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 385–394.
[14] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representations (ICLR)*, Feb. 2017, pp. 1–14.
[15] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Feb. 2018, pp. 1–12.
[16] H.-P. Hsieh, R. Yan, and C.-T. Li, "Where you go reveals who you know: Analyzing social ties from millions of footprints," in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, Oct. 2015, pp. 1839–1842.
[17] C. Wang, J. Zhou, and B. Yang, "From footprint to friendship: Modeling user followership in mobile social networks from check-in data," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Aug. 2017, pp. 825–828.
[18] B. Aronov *et al.*, "Are friends of my friends too social? Limitations of location privacy in a socially-connected world," in *Proc. 19th Int. Symp. Mobile Ad Hoc Netw. Comput.*, Jun. 2018, pp. 280–289.
[19] F. Zhou, B. Wu, Y. Yang, G. Trajcevski, K. Zhang, and T. Zhong, "Vec2Link: Unifying heterogeneous data for social link prediction," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, Oct. 2018, pp. 1843–1846.
[20] H.-P. Hsieh and C.-T. Li, "Inferring online social ties from offline geographical activities," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, p. 17, Jan. 2019.
[21] H. Cai, V. W. Zheng, and K. C.-C. Chang, "A comprehensive survey of graph embedding: Problems, techniques, and applications," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 9, pp. 1616–1637, Sep. 2018.
[22] M. Backes, M. Humbert, J. Pang, and Y. Zhang, "Walk2Friends: Inferring social links from mobility profiles," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, Oct. 2017, pp. 1943–1957.
[23] Q. Gao, G. Trajcevski, F. Zhou, K. Zhang, T. Zhong, and F. Zhang, "Trajectory-based social circle inference," in *Proc. ACM SIGSPATIAL Int. Conf. Adv. Geograph. Inf. Syst.*, Nov. 2018, pp. 369–378.
[24] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proc. Int. Conf. Learn. Representations (ICLR)*, Jan. 2013, pp. 1–12.
[25] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, May 2014, pp. 1–14.
[26] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
[27] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. (2014). "Empirical evaluation of gated recurrent neural networks on sequence modeling." [Online]. Available: https://arxiv.org/abs/1412.3555
[28] A. Bojchevski and S. Günnemann, "Deep Gaussian embedding of graphs: Unsupervised inductive learning via ranking," in *Proc. Int. Conf. Learn. Representations (ICLR)*, Feb. 2018, pp. 1–13.
[29] J. Zhou *et al.* (2018). "Graph neural networks: A review of methods and applications." [Online]. Available: https://arxiv.org/abs/1812.08434
[30] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu. (2019). "A comprehensive survey on graph neural networks." [Online]. Available: https://arxiv.org/abs/1901.00596
[31] Z. Li, X. Fang, and O. Sheng, "A survey of link recommendation for social networks: Methods, theoretical foundations, and future research directions," *ACM Trans. Manage. Inf. Syst.*, vol. 9, no. 1, p. 1, Feb. 2018.
[32] M. De Choudhury, W. A. Mason, J. M. Hofman, and D. J. Watts, "Inferring relevant social networks from interpersonal communication," in *Proc. 19th Int. Conf. World Wide Web*, Apr. 2010, pp. 301–310.
[33] J. Leskovec, D. Huttenlocher, and J. Kleinberg, "Predicting positive and negative links in online social networks," in *Proc. 19th Int. Conf. World Wide Web*, pp. 641–650, Apr. 2010.
[34] D. Wang, D. Pedreschi, C. Song, F. Giannotti, and A.-L. Barabási, "Human mobility, social ties, and link prediction," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2011, pp. 1100–1108.
[35] P. Wang, F. Sun, D. Wang, J. Tao, X. Guan, and A. Bifet, "Inferring demographics and social networks of mobile device users on campus from AP-trajectories," in *Proc. 26th Int. Conf. World Wide Web Companion*, Apr. 2017, pp. 139–147.
[36] H. Wang, Z. Li, and W.-C. Lee, "PGT: Measuring mobility relationship using personal, global and temporal factors," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Dec. 2014, pp. 570–579.
[37] B. Alharbi, A. Qahtan, and X. Zhang, "Minimizing user involvement for learning human mobility patterns from location traces," in *Proc. AAAI Conf. Artif. Intell.*, Feb. 2016, pp. 865–871.
[38] G. Yang and A. Züfle, "Spatio-temporal prediction of social connections," in *Proc. 4th Int. ACM Workshop Manage. Mining Enriched Geo-Spatial Data*, New York, NY, USA, May 2017, p. 6.
[39] I. Psorakis, S. J. Roberts, I. Rezek, and B. C. Sheldon, "Inferring social network structure in ecological systems from spatio-temporal data streams," *J. Roy. Soc. Interface*, vol. 9, no. 76, pp. 3055–3066, Jun. 2012.
[40] D. Brockmann, L. Hufnagel, and T. Geisel, "The scaling laws of human travel," *Nature*, vol. 439, pp. 462–465, Jan. 2006.
[41] M. C. González, C. A. Hidalgo, and A.-L. Barabási, "Understanding individual human mobility patterns," *Nature*, vol. 453, pp. 779–782, Jun. 2008.
[42] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási, "Limits of predictability in human mobility," *Science*, vol. 327, no. 5968, pp. 1018–1021, Feb. 2010.
[43] F. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi, "Trajectory pattern mining," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2007, pp. 330–339.
[44] F. Giannotti *et al.*, "Unveiling the complexity of human mobility by querying and mining massive trajectory data," *VLDB J.–Int. J. Very Large Data Bases*, vol. 20, no. 5, pp. 695–719, Oct. 2011.
[45] J. J.-C. Ying, W.-C. Lee, and V. S. Tseng, "Mining geographic-temporal-semantic patterns in trajectories for location prediction," *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 1, pp. 1–34, Dec. 2013.
[46] J. Feng *et al.*, "Deepmove: Predicting human mobility with attentional recurrent networks," in *Proc. Int. Conf. World Wide Web*, Apr. 2018, pp. 1459–1468.
[47] Q. Gao, F. Zhou, G. Trajcevski, K. Zhang, T. Zhong, and F. Zhang, "Predicting human mobility via variational attention," in *Proc. Int. Conf. World Wide Web (WWW)*, May 2019, pp. 1452–1458.
[48] Q. Liu, S. Wu, L. Wang, and T. Tan, "Predicting the next location: A recurrent model with spatial and temporal contexts," in *Proc. 13th AAAI Conf. Artif. Intell.*, Feb. 2016, pp. 194–200.
[49] F. Zhou, R. Yin, K. Zhang, G. Trajcevski, and T. Zhong, "Adversarial point-of-interest recommendation," in *Proc. Int. Conf. World Wide Web (WWW)*, May 2019, pp. 1257–1263.
[50] C. Ma, Y. Zhang, Q. Wang, and X. Liu, "Point-of-interest recommendation: Exploiting self-attentive autoencoders with neighbor-aware influence," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, Oct. 2018, pp. 697–706.
[51] Z.-H. Zhou and M.-L. Zhang, *Multi-Label Learning*. Boston, MA, USA: Springer, 2017.
[52] J. Feng and Z.-H. Zhou, "Deep MIML network," in *Proc. AAAI Conf. Artif. Intell.*, Feb. 2017, pp. 1884–1890.

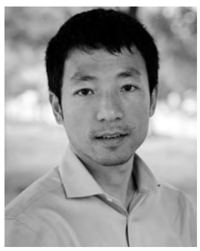
[53] M. Peters *et al.*, "Deep contextualized word representations," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, Jun. 2018, pp. 2227–2237.
[54] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 5998–6008.
[55] Z. Lin *et al.*, "A structured self-attentive sentence embedding," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Mar. 2017, pp. 1–15.
[56] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–15.
[57] P. Yang, X. Sun, W. Li, S. Ma, W. Wu, and H. Wang, "SGM: Sequence generation model for multi-label classification," in *Proc. 27th Int. Conf. Comput. Linguistics (COLING)*, Jun. 2018, pp. 3915–3926.
[58] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: User movement in location-based social networks," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2011, pp. 1082–1090.
[59] D. Yang, D. Zhang, L. Chen, and B. Qu, "NationTelescope: Monitoring and visualizing large-scale collective behavior in LBSNs," *J. Netw. Comput. Appl.*, vol. 55, pp. 170–180, Sep. 2015.
[60] K. H. Lim, J. Chan, C. Leckie, and S. Karunasekera, "Personalized tour recommendation based on user interests and points of interest visit durations," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, Jun. 2015, pp. 1778–1784.
[61] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *Mach. Learn.*, vol. 85, no. 3, p. 333, Dec. 2011.
[62] Z.-H. Zhou, M.-L. Zhang, S.-J. Huang, and Y.-F. Li, "Multi-instance multi-label learning," *Artif. Intell.*, vol. 176, no. 1, pp. 2291–2320, Jan. 2012.

TING ZHONG received the B.S. degree in computer application and the M.S. degree in computer software and theory from Beijing Normal University, Beijing, China, respectively, in 1999 and 2002, respectively, and the Ph.D. degree in information and communication engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2009, where she was a Lecturer (2003–2009) and has been an Associate Professor, since 2010. Her research interests include deep learning, social networks, and cloud computing.

FANG LIU received the B.S. degree in computer science and engineering from Northwest Normal University, Lanzhou, China, in 2017. She is currently pursuing the M.S. degree with the University of Electronic Science and Technology of China. Her current research interests include recommendation systems, spatio-temporal data mining, and deep generative learning.

FAN ZHOU received the B.S. degree in computer science from Sichuan University, China, in 2003, and the M.S. and Ph.D. degrees from the University of Electronic Science and Technology of China, in 2006 and 2011, respectively, where he is currently an Associate Professor with the School of Information and Software Engineering. His research interests include machine learning, spatio-temporal data management, and social network knowledge discovery.

GOCE TRAJCEVSKI received the B.Sc. degree from the University of Sts. Kiril i Metodij, and the M.S. and Ph.D. degrees from the University of Illinois at Chicago. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, Iowa State University. His main research interests are in the areas of spatio-temporal data management, uncertainty and reactive behavior management in different application settings, and incorporating multiple contexts. In addition to a book chapter and three encyclopedia chapters, he has coauthored over 110 publications in refereed conferences and journals, and has received two Best Paper Awards (CoopIS 2000 and IEEE MDM 2010), the Best Short-Paper Award (ACM MSWiM 2013), and the Best Demo Award (SSTD 2017). His research has been funded by the NSF, ONR, BEA, and Northrop Grumman Corp. He was the General Co-Chair of the IEEE ICDE 2014, the PC Co-Chair of the ADBIS 2018 and ACM SIGSPATIAL 2016 and 2017, and has served in various roles in organizing committees in numerous conferences and workshops. He is an Associate Editor of the ACM TSAS and the *Geoinformatica Journals*.



KUNPENG ZHANG received the Ph.D. degree in computer science from Northwestern University. He is a Researcher in the area of large-scale data analysis, with particular focuses on mining social media data through machine learning, network analysis, and natural language processing techniques. He is currently an Assistant Professor with the Department of Information Systems, Smith School of Business, University of Maryland, College Park, MA, USA. He has published papers in the area of social media, text mining, network analysis, and information systems on top conference and journals. He serves as program committees for many international conferences and is currently an Associate Editor for the *Electronic Commerce Research Journal*.