NCF: A Neural Context Fusion Approach to Raw Mobility Annotation

Renjun Hu, Jingbo Zhou, Xinjiang Lu, Hengshu Zhu, Shuai Ma, and Hui Xiong

Abstract—Understanding human mobility patterns at the point-of-interest (POI) scale plays an important role in enhancing business intelligence in mobile environments. While large efforts have been made in this direction, most studies simply utilize POI check-ins to mine the concerned mobility patterns, the effectiveness of which is usually hindered due to data sparsity. To obtain better POI-based human mobility for mining, in this paper, we strive to directly annotate the POIs associated with raw user-generated mobility records. We propose a neural context fusion approach which integrates various context factors in people's POI-visiting behaviors. Our approach evaluates the preference and transition factors via representation learning. Notably, we incorporate an attention mechanism to deal with the randomized transitions in raw mobility. The domain knowledge factors, i.e., distance, time and popularity, remain effective and our approach further includes them from a data-driven perspective. Factors are automatically fused with a feed-forward neural network. Furthermore, we exploit a multi-head architecture to enhance the model expressiveness. Using two real-life data sets, we conduct our experimental study and find that our approach consistently outperforms the state-of-the-art baselines by at least 32% in accuracy. Besides, we demonstrate the utility of the obtained POI-based human mobility with a POI recommendation example.

Index Terms—Raw mobility annotation; Point-of-Interest; neural network; business intelligence

1 Introduction

Human mobility is the movements of human beings in space and time [1]. Based on people's diverse emphases of movements, human mobility can be roughly classified into three categories: location-based, activity-based and point-of-interest (POI) based. The location-based focuses on the space and time aspects [1], [2], [3], [4], and the activity-based essentially explains the purposes behind people's moves [5], [6], [7], [8], both of them have greatly enhanced our understanding of urban dynamics. Besides the above two, effort has also been paid to the acquisition of POI-based human mobility [9], [10], [11], i.e., people's movements between POIs. Intuitively, POIs contain rich semantics and play an important role in mobile services and business intelligence. Thus, POI-based human mobility not only is essential for mobility-related applications, e.g., weekly mobility pattern identification [12], city planning [13], epidemic diffusion analysis [14] and resource allocation [15] but also remains fundamental in POI-centric tasks, such as POI and trajectory recommendations [16], [17] and POI demand forecasting [18].

Check-in records are a good and off-the-shelf source of POI-based human mobility [19], [20], [21], [17]. However, they are sparse by nature, from which the downstream applications may suffer. For instance, the Foursquare data set collected by [22] contains 227,428 check-ins of 1,083 users in a span of ten months in New York City. Note that inactive users who have less than 3 check-ins per week have already been filtered out. Even though, each user still only has 0.675 check-in per day on average, which is far from enough for recording our daily movements. Worse still, Wu and Li [10] experimentally verifies that temporally sparse mobility may not exhibit any significant transitional relationships. These results affect the basis of sequential modeling for human mobility recorded by check-ins.

To obtain better POI-based human mobility, in this paper we study raw mobility annotation, aiming to *directly annotate the raw mobility records (i.e., user-generated timestamped locations) with associated POIs*. Observe that: (i) raw mobility data can be collected with reasonable user participation by various devices running some mobile services, e.g., Google Maps or Yelp, and (ii) for any POI check-in, a raw mobility record can be collected by simply ignoring the POI. In this sense, raw mobility data is much easier to collect and can track more time-resolved individual locations compared with check-ins. Moreover, direct annotation could help to alleviate the noise introduced by fake check-ins [23]. Therefore, proactive acquisition of POI-based human mobility from user-generated timestamped locations can provide both densely-sampled trajectories and reliable semantics to support numerous potential applications.

This task also remains challenging since individuals' POI-visiting behaviors are influenced by various complex contexts and turn out to be more stochastic. More specifically, raw mobility records are usually more adequate than check-ins. This enables and requires a better exploitation of personal preferences for visiting POIs. The work in [9] simply counts the number of previous visits from a user at a POI

- *R. Hu and S. Ma are with the SKLSDE Lab, Beihang University and the Beijing Advanced Innovation Center for Big Data and Brain Computing, Beijing, China. E-mail: {hurenjun, mashuai}@buaa.edu.cn.*
- *J. Zhou and X. Lu are with the Business Intelligence Lab, Baidu Research, National Engineering Laboratory of Deep Learning Technology and Application; H. Zhu is with the Talent Intelligence Center, Baidu Inc., Beijing, China. E-mail: {zhoujingbo, luxinjiang, zhuhengshu}@baidu.com.*
- *H. Xiong is with the Management Science and Information Systems Department, Rutgers Business School, Rutgers University, Newark, NJ, USA. E-mail: hxiong@rutgers.edu.*

Manuscript received July, 2019; revised December, 2019 and March, 2020.

to model preference, which might suffer greatly from data sparsity. While [10] proposes to enforce category-level consistency in spatially- and temporally-close mobility records, which is a very strong assumption. Thus, these strategies are insufficient to support raw mobility annotation. Also observe that preferences for POIs are mutually influenced between people, which has been largely ignored in earlier work. Second, transitional patterns, i.e., sequential patterns of movements, are very likely to exhibit certain randomness in raw human mobility, instead of being strictly sequential. For instance, people might visit other POIs within their routine transitional patterns. Such randomized effect should be considered. Last but not the least, the influence of distance, time and POI popularity ought to be properly incorporated. These basic factors play a decisive role in our daily POI-visiting behaviors.

However, none of existing solutions [9], [10], [11] integrate all the above visiting contexts. Worse still, they model the complex contexts in simple ways, ignore the randomized effect of raw mobility, and combine contexts in pre-defined manners. To this end, we propose a Neural Context Fusion approach, namely NCF, to tackling raw mobility annotation. It enables to capture the complex preference and transition structures at the fine-grained user, POI, and region levels. Moreover, it learns to fuse various key context factors in POI-visiting behaviors in an end-to-end data-driven manner. To the best of our knowledge, NCF is the first neural model for mobility annotation.

To be specific, NCF first derives the preference factor via representation learning (RL). RL refers to the technique of embedding data points into low-dimensional hidden spaces. It can effectively reveal the hidden structures in original data, e.g., the preference relationships between user and POI, and has already been exploited for human mobility analysis [17], [19], [24]. For each user or POI, NCF learns a vector in a hidden space such that a pair of user and POI vectors are placed close if the user has preference for the POI. By this, user vectors can encode people's distinct preferences for POIs. As a side effect, RL also provides a natural way for mutual preference influence through vector manipulations in the hidden space.

From a transitional point of view, we notice that the previous mobility records also offer clues for mobility annotation, i.e., how likely someone visits a POI given the places she/he stays earlier. Since raw mobility data only track locations, we assign locations to road-segmented regions for learning transitional patterns. The reasons are two-fold. First, regions can be easily incorporated in the RL framework. Second, the functionality of regions can well encode transitional patterns, e.g., *residence-to-work*. Regions are also embedded in the same hidden space as users and POIs. Based on region vectors, NCF derives two transition factors from the POI and user perspectives, respectively. Also recall that transitions in raw mobility are randomized. In other words, each of the previous regions may have a direct impact on the current visited POI. We adopt an attention mechanism [25] to capture such randomness. It automatically determines an importance weight of each region and computes a weighted sum of region representations, which ensures the direct impact of each region.

The remaining distance, time and popularity factors are inspired by domain knowledge, i.e., people tend to visit POIs that are close to their locations, active at the visit time and popular among people. These three factors are data-driven in our approach. Inspired by data statistics, we adopt an exponentially-decayed function to evaluate the distance factor. While the time and popularity factors are estimated from map search query data, i.e., timestamped search logs from users to POIs on map services, due to the better POI coverage compared with POI visit data.

Table 1
Descriptions of mathematical notations

Notation	Description
$x = (l, t)$	Mobility record of location l and time stamp t
$\mathcal{T}_u$	Trajectory of user u, $\mathcal{T}_u = [x_1, x_2, \dots, x_L]$
$\mathcal{T}_u[i, j]$	Sub-trajectory $[x_i, \dots, x_j]$ of $\mathcal{T}_u$
ξ_D, ξ_T, ξ_S	Distance, temporal closeness and spanning thresholds
$\mathcal{R}_t^u$	Set of regions user u visits before time stamp t
H, h	Total number of heads and a specific head
u, $\mathbf{u}^{(h)}$	User and user embedding in the h-th head
p, $\mathbf{p}^{(h)}$	POI and POI embedding in the h-th head
r, $\mathbf{r}^{(h)}$	Region and region embedding in the h-th head
$\mathsf{MLP}^{(h)}(\cdot)$	Multilayer perceptron function in the h-th head
$a^{(h)}(\cdot, \cdot)$	Attention mechanism in the h-th head

All the above factors are fed into a feed-forward fusion neural network to compute the visit probabilities of candidate POIs. Moreover, NCF exploits a multi-head architecture such that each head evaluates an independent set of RL factors. Such an architecture is believed to enhance the expressive power of NCF, as it can learn a distinct POI-visiting pattern in each head.

To sum up, our main contributions are as follows:

(1) We investigate the raw mobility annotation problem to overcome the limitation of POI check-in data.
(2) We propose a neural context fusion (NCF) approach. It integrates various key context factors in people's POI-visiting behaviors and is equipped with a context fusion neural network and a multi-head architecture.
(3) We evaluate NCF using two real-life data sets. We find that NCF significantly outperforms the state-of-the-art baselines by at least 32% in accuracy and remains efficient. Moreover, both the context factors and the multi-head architecture enhance the effectiveness.
(4) We demonstrate the utility of POI-based human mobility with a POI recommendation example. We show that, even using a very simple recommendation strategy, the annotated POIs by NCF can substantially promote the recommendation accuracy by at least 11.2%.

The rest of the paper is organized as follows. Section 2 introduces raw mobility preprocessing and formalizes the problem. Our NCF approach is described in Section 3. We present the experimental study in Section 4, followed by related work in Section 5 and concluding remarks in Section 6.

2 Raw Mobility Preprocessing

Raw human mobility cannot be directly annotated as a fraction of records are produced when people are in moving status. In this section, we introduce how to preprocess raw human mobility for annotation and formalize our problem. The notations used in this paper are listed in Table 1.

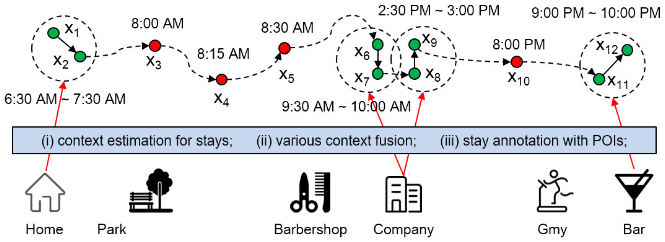


Figure 1. An example of raw mobility annotation, where points in green and red denote staying and moving mobility records, respectively, and each dashed circle represents a stay.

Let a mobility record $x = (l, t)$ be a pair of location l (latitude and longitude) and time stamp t. Let $\mathcal{U}$ and $\mathcal{P}$ be the sets of users and POIs, respectively.

Definition 1 (Trajectory). A trajectory $\mathcal{T}_u$ of user $u \in \mathcal{U}$ is a sequence of ordered mobility records generated by user u, i.e., $\mathcal{T}_u = [x_1, x_2, \ldots, x_L]$, where L is the length of $\mathcal{T}_u$ and the time stamps satisfy $t_1 < t_2 < \cdots < t_L$.

We introduce a notion of stay to distinguish between mobility records that can or cannot be annotated with POIs. Let $\mathcal{T}_u[i,j] = [x_i, \ldots, x_j]$ $(1 \leq i < j \leq L)$ denote a sub-trajectory of $\mathcal{T}_u$. And we say a sub-trajectory is a stay if all mobility records included are spatially and temporally close and last long enough. In this case, the corresponding mobility records are very likely to be associated with a POI where the user visits when generating these records.

Definition 2 (Stay). Given distance, temporal closeness and spanning thresholds ξ_D, ξ_T and ξ_S, sub-trajectory $\mathcal{T}_u[i,j]$ is a stay if (a) $\mathsf{dist}(l_k, \bar{l}_{ij}) \leq \xi_D$ holds for all $i \leq k \leq j$, where $\bar{l}_{ij}$ is the mean of $\{l_i, \ldots, l_j\}$, (b) $t_{k+1} - t_k \leq \xi_T$ holds for all $i \leq k \leq j-1$, and (c) $t_j - t_i \geq \xi_S$.

Intuitively, distance threshold ξ_D provides tolerance for positioning errors of daily use devices, temporal closeness threshold ξ_T is the typical amount of time within which people can complete a visit to a POI, and, finally, spanning threshold ξ_S is the amount of time that we can assure a POI visit by someone with high confidence. We next illustrate these concepts with an example.

Example 1. Figure 1 gives a trajectory $\mathcal{T}_u$ of user u collected in one day. Assume the diameter of circles in Fig. 1 is 200 meters, $\xi_D = 200$ meters, $\xi_T = 2$ hours and $\xi_S = 10$ minutes. We have the following.

- Sub-trajectories $\mathcal{T}_u[1,2]$ and $\mathcal{T}_u[11,12]$ are stays since the included mobility records are spatially and temporally close and span longer than the spanning threshold.
- Despite the distance and spanning time, x_6–x_9 belong to two stays $\mathcal{T}_u[6,7]$ and $\mathcal{T}_u[8,9]$ since x_7 and x_8 are not temporally close (i.e., $t_8 - t_7 > \xi_T$). Note that user u might visit other POIs during 10:00 AM and 2:30 PM.
- Spatiotemporal points x_3, x_4, x_5 and x_{10} are moving mobility records. They do not belong to any stays due to the distance constraint.

Similar to the stay point detection algorithm in [26], we can identify a set of non-overlapping stays from each $\mathcal{T}_u$ in $O(L)$ time. The main idea is to enumerate a mobility record (say x_i) and find the longest sub-trajectory $\mathcal{T}_u[i,j]$ satisfying the conditions of a stay. If that sub-trajectory does not exist, turn to x_{i+1}. Otherwise, identify $\mathcal{T}_u[i,j]$ as a stay and start with x_{j+1}. It is easy to verify that the above process only needs to scan through the trajectory once.

In most cases, the identified stays are associated with POIs. For instance, $\mathcal{T}_u[1,2]$ in Fig. 1 is generated when u is at home, $\mathcal{T}_u[6,7]$ and $\mathcal{T}_u[8,9]$ are annotated with the company, and $\mathcal{T}_u[11,12]$ corresponds to a bar. We thus (i) estimate various context of stays and further (ii) fuse these context factors to (iii) guide stay annotation with the corresponding POIs (the details will be described in Section 3). We then obtain the POI-based human mobility, which can facilitate a broad range of applications. It is noteworthy that, for a specific setting of the thresholds, the POIs that are annotated to mobility records might have some bias, e.g., convenience stores are probably excluded if ξ_S is set to 10 minutes. Formally, our problem is as below.

Problem 1 (Raw Mobility Annotation). Given a trajectory $\mathcal{T}_u$, raw mobility annotation is to identify a set $\mathcal{S} = \{\mathcal{T}_u[i,j]\}$ of stays from $\mathcal{T}_u$ and annotate a POI $p \in \mathcal{P}$ to each $\mathcal{T}_u[i,j]$ such that user u visits POI p when generating the mobility records included in $\mathcal{T}_u[i,j]$.

3 A Neural Context Fusion Approach

In this section, we introduce our neural context fusion (NCF) approach to tacking the problem. We first present the framework overview, then illustrate the exploited context factors and, finally, specify the training and inference procedures.

3.1 Framework Overview

Our NCF attacks the annotation problem by evaluating the visit probability of each POI near the stay location. Once all nearby POIs are processed, the POI with the highest visit probability is then annotated to the stay. Unlike crowd-level human mobility that has universal governing rules [3], individuals' mobile behaviors are influenced by various complex contexts and turn out to be more stochastic. Accordingly, NCF integrates various context factors to estimate visit probabilities. The framework overview of NCF is illustrated in Fig. 2. It takes as input (a) a user $u \in \mathcal{U}$, (b) the location l and time stamp t of a stay $\mathcal{T}_u[i,j]$ (where l is the mean of $\{l_i, \ldots, l_j\}$ and $t = (t_i + t_j)/2$), (c) a set $\mathcal{R}_u^t$ of regions where user u stays before time stamp t, and (d) a candidate POI $p \in \mathcal{P}$, and outputs the probability that u is visiting p when staying at location l and time stamp t.

Our NCF first looks up the embeddings of user u, candidate POI p and regions $r \in \mathcal{R}_u^t$. Based on the candidate POI p, the region embeddings are aggregated together with an attention network. It then derives three representation learning (RL) factors via evaluating the pairwise inner products of the user, POI and aggregated region embeddings. To obtain better expressive power, a multi-head architecture [27] is further exploited in NCF. Each head computes an independent set of RL factors. Such process is repeated for H times, resulting in a total of $3H$ RL factors. By doing so, NCF can learn a distinct visiting pattern in each head. In addition, NCF incorporates another three data-driven domain knowledge factors. Finally, with a feed-forward fusion network, NCF learns to automatically fuse these factors to compute the visit probability of a POI given the stay.

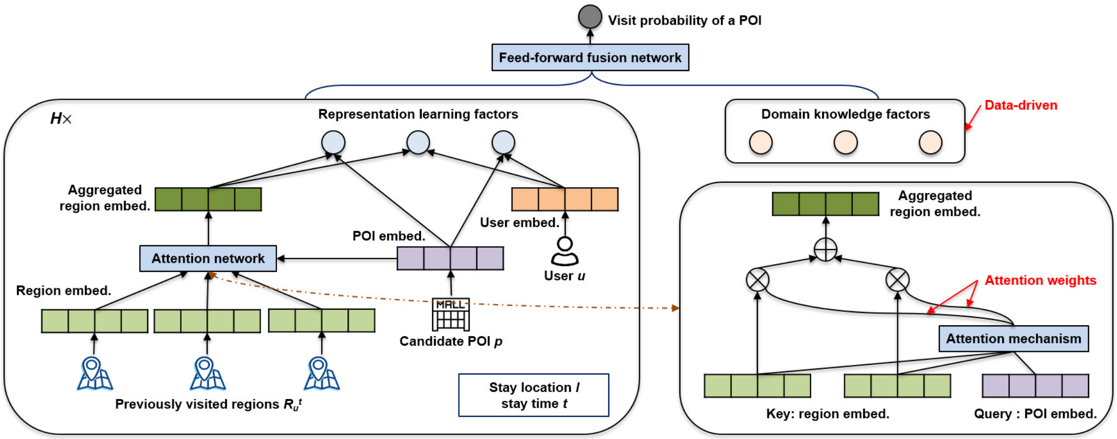


Figure 2. Framework overview of NCF

3.2 Representation Learning Factors

We next explain the details of deriving RL factors in the h-th ($1 \leq h \leq H$) head. Note that all operations are the same across different heads except that each head learns its own embedding and model parameters.

The first RL factor evaluates the preference between users and POIs. Assume the dimensionality of NCF is d. Basically, NCF embeds all users and POIs in a hidden space $\mathbb{R}^d$ and computes the preference of a user to a POI as the inner product of the corresponding user and POI embeddings. Formally, the preference factor $F_{pref}^{(h)}(u,p)$ of user u and candidate POI p in the h-head is as follows:

$$F_{pref}^{(h)}(u,p) = \mathsf{MLP}^{(h)}(\mathbf{u}^{(h)})^T \mathsf{MLP}^{(h)}(\mathbf{p}^{(h)}), \tag{1}$$

where $\mathbf{u}^{(h)}$ and $\mathbf{p}^{(h)}$ are the embeddings of u and p in the h-th head of NCF, respectively, and $\mathsf{MLP}^{(h)}$ is a head-specific multilayer perceptron (MLP) that performs a non-linear transformation. The MLP is introduced for two reasons: (a) to distinguish between embeddings used for evaluating context factors here and evaluating attention weights later, and (b) to prevent our model from overfitting via leveraging an MLP in conjunction with dropout [28]. We adopt an MLP with two layers of adaptive weights:

$$\mathsf{MLP}^{(h)}(\mathbf{x}) = \tanh(\mathbf{W}_2^{(h)} \tanh(\mathbf{W}_1^{(h)}\mathbf{x} + \mathbf{b}_1^{(h)}) + \mathbf{b}_2^{(h)}). \tag{2}$$

$\mathsf{MLP}^{(h)}$ learns two adaptive weight matrices $\mathbf{W}_1^{(h)}, \mathbf{W}_2^{(h)} \in \mathbb{R}^{d\times d}$ and two bias vectors $\mathbf{b}_1^{(h)}, \mathbf{b}_2^{(h)} \in \mathbb{R}^d$. That is, the transformation is shared for all users and POIs in the same heads, while different heads exploit different transformation parameterized by their own matrices and biases.

In addition to users and POIs, NCF also embeds regions in the hidden representation space. Regions, especially those road-segmented ones, usually exhibit certain functionalities that meet people's different needs of socioeconomic activities [29]. In this sense, they also provide clues from the transitional perspective for raw mobility annotation. For instance, a person is in general more likely to visit an office building than a hotel if she/he was found in a residential area earlier. By further learning region representations, NCF incorporates the user- and POI-based transitional relationships that evaluate how likely user u transfers from previous regions to the candidate POI.

Recall that $\mathcal{R}_u^t$ is the set of regions visited by user u before time stamp t. From the transitional perspective, a candidate POI p is likely to be visited if the transition rates from regions $r \in \mathcal{R}_u^t$ to p are high. The simplest way to achieve this is to evaluate the average inner product between region embeddings $\mathbf{r}$ of $r \in \mathcal{R}_u^t$ and POI embedding $\mathbf{p}$. In this way, each region is assigned an equal weight for POI p, which is usually not the true case in practice. Alternatively, we can derive a sequence of regions based on the stay time at regions $r \in \mathcal{R}_u$. This can lead to a sequential model assuming that user u visits these regions sequentially and, because of that, finally visits POI p. However, sequential modeling is too restrict for raw mobility analysis, both due to the incompleteness of the recorded raw mobility data and the randomized effect in human mobility.

To address the above two issues, our NCF approach adopts an attention mechanism, as shown in the lower right of Fig. 2. An attention function takes a query and a set of values as input and outputs a query-aware weighted sum of the values [27]. By treating candidate POI p as the query and regions in $\mathcal{R}_u^t$ as values, the attention function lets POI p decide the regions from which the transition starts. Formally, we assign each region $r \in \mathcal{R}_u^t$ a distinct weight automatically determined by candidate POI p and region r themselves. Afterward, we compute an aggregated region embedding $\hat{\mathbf{r}}_u^t$ as the attention output:

$$\hat{\mathbf{r}}_u^{t(h)} = \sum_{r\in\mathcal{R}_u^t} \frac{a^{(h)}(\mathbf{p}^{(h)}, \mathbf{r}^{(h)})}{\sum_{r'\in\mathcal{R}_u^t} a^{(h)}(\mathbf{p}^{(h)}, \mathbf{r'}^{(h)})} \mathbf{r}^{(h)}. \tag{3}$$

Here $a^{(h)}(,)$ is the attention mechanism in the h-th head which determines the weights of $\mathbf{r}^{(h)}$. These weights indicate the importance of regions to POI p. In other words, the attention mechanism decides to which regions POI p should pay attention. In this study, we adopt the simple dot-product attention mechanism [27]. It first performs a shared transformation on POI and region embeddings and then computes their dot-product as the attention weight:

$$a^{(h)}(\mathbf{p}^{(h)}, \mathbf{r}^{(h)}) = \mathsf{MLP}_{\mathsf{attn}}^{(h)}(\mathbf{p}^{(h)})^{\mathsf{T}} \mathsf{MLP}_{\mathsf{attn}}^{(h)}(\mathbf{r}^{(h)}). \tag{4}$$

Note that $\mathsf{MLP}_{\mathsf{attn}}^{(h)}$ is another head-specific 2-layer MLP, which is the same to $\mathsf{MLP}^{(h)}$ in Eq. (2) except for adopting a ReLU (Rectified Linear Unit) nonlinearity and learning its

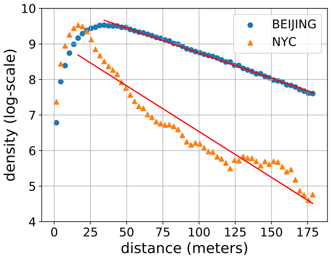


Figure 3. Distribution of distance between stay locations and POIs

own parameters. ReLU$(x) = \max(0, x)$ is defined as the positive part of its argument and has been demonstrated to enable better training of deep networks. In this way, MLP$^{(h)}$ and MLP$^{(h)}_{\text{attn}}$ can distinguish between embeddings used for evaluating context factors and attention weights.

We then derive the *POI-based transition* context factor $F^{(h)}_{pbtr}(\mathcal{R}^t_u, p)$ as the inner product between the aggregated region embedding and the POI embedding:

$$F^{(h)}_{pbtr}(\mathcal{R}^t_u, p) = \mathsf{MLP}^{(h)}(\hat{\mathbf{r}}^{t(h)}_u)^\mathsf{T}\mathsf{MLP}^{(h)}(\mathbf{p}^{(h)}). \tag{5}$$

This can be interpreted in a way that each region performs a direct transition to POI p and the overall region-POI transition is a weighted sum of these direct transitions.

Similarly, we further derive the *user-based transition* context factor $F^{(h)}_{ubtr}(\mathcal{R}^t_u, u)$ based on the aggregated region embedding and user embedding:

$$F^{(h)}_{ubtr}(\mathcal{R}^t_u, u) = \mathsf{MLP}^{(h)}(\hat{\mathbf{r}}^{t(h)}_u)^\mathsf{T}\mathsf{MLP}^{(h)}(\mathbf{u}^{(h)}). \tag{6}$$

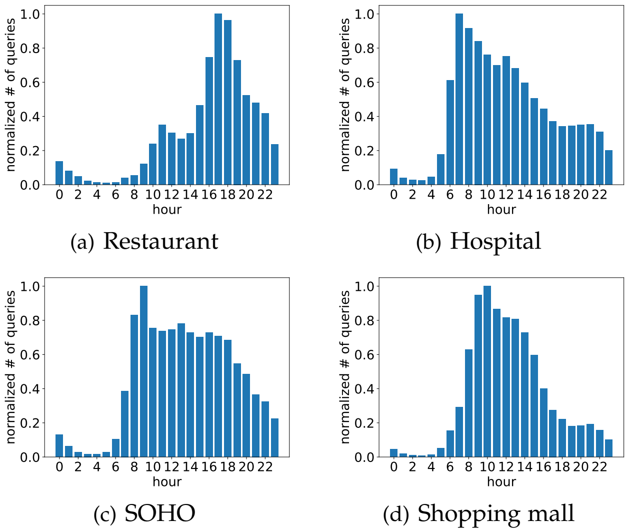
Note that "attentioned" by candidate POI p, the aggregated region embedding is mainly contributed by regions that have high transition rates to POI p. Here the user-based transition factor verifies the weighted transition from user perspective, i.e., whether user u is highly-related to these contributing regions of POI p.

Remarks. (1) We have tried sequential modeling for NCF by equipping a positional encoder in the attention [27], and found it rarely leaded to improvement. This indicates that strictly sequential modeling is not necessary for raw mobility and, hence, verifies the randomized effect for raw human mobility. For the sake of simplicity, we only consider the direct transitions from earlier regions. (2) Technically, NCF can take any number of regions around the annotated one as input. However, many real-life applications (e.g., recommendation) require to annotate mobility records once they are generated, which means there only exist previous regions. Besides, it is desired to keep the inference component light-weight and avoid introducing noises by unrelated regions. Hence, we finally choose to use a small number, e.g., 10, of the most recent previous regions as $\mathcal{R}^t_u$.

3.3 Domain Knowledge Factors

The remaining three context factors in NCF are inspired by domain knowledge. Essentially, these factors give biases to POIs that are close to location l, active at time stamp t and popular among people in general. We exploit a data-driven strategy to estimate these factors.

First, the *distance* context factor $F_{dist}(p, l)$ captures the spatial preference that people are more likely to visit POIs nearby. It is usually achieved by an exponentially decayed function w.r.t. distance. However, the decay rate varies in different formulas, e.g., linear in [10] and squared in RBF kernel [30]. To determine an appropriate formula, we collect the statistics of distance between people's stay locations and visited POIs on our BEIJING and NYC data (refer to Section 4 for data set details). The density distribution is reported in Fig. 3. Note that the y-axis is log-scaled and there is an obvious linear correlation between the distance and the density on both data sets. We hence adopt an exponential function with a negative exponent weight for linear distance, the same to the node potential in [10]:



(a) Restaurant

(b) Hospital

(c) SOHO

(d) Shopping mall

Figure 4. The number of hourly map queries for POI activeness

$$F_{dist}(p, l) = \exp(-\phi \cdot \mathsf{dist}(l, p)), \tag{7}$$

where $\mathsf{dist}(l, p)$ is the distance (in meter) between stay location l and POI p and $\phi > 0$ is a decaying parameter. A relatively low ϕ is preferred if the positioning accuracy is low and vice versa. In practice it suffices to choose a ϕ within [0.001, 0.05]: $F_{dist}(p, l)$ halves every (693, 14) meters with $\phi = (0.001, 0.05)$, respectively.

Similarly, the *time* factor captures the temporal preference that users are more likely to visit POIs that are active at the stay time. This calls for a metric to evaluate the activeness of POIs at different time. Basically, we can use the number of visits. However, large-scale visit record data is usually hard to obtain, and check-ins are highly biased to certain types of POIs such as scenic spots. On the other hand, obtaining map search query data is much easier, and there is a strong correlation between map queries and POI visits [31]. To further verify this, we collect and report the map search query statistics of four typical POIs in Fig. 4. As shown, such map search query statistics are a good proxy of POI activeness. Notably, the selected restaurant is famous for its supper, and its highest activeness in the evening is also well reflected by the map search query data in Fig. 4(a). The peak time of the hospital, SOHO and shopping mall occurs at 7 AM, 9 AM and 10 AM, respectively, which also conforms to real-life cases. Hence, we use the number of map search queries to evaluate the POI activeness. More specifically, let $Q_{p,k}$ denote the number of map queries of

POI p in time slot $k \in \{1, \dots, T\}$ of a day. The corresponding activeness $A_{p,k}$ is then estimated by smoothing $Q_{p,k}$ with a log operator and a constant bias and normalizing:

$$A_{p,k} = \begin{cases} 0, & \text{if } \max_{k'} Q_{p,k'} = 0; \\ \frac{\log(Q_{p,k}+1)}{\log(\max_{k'} Q_{p,k'}+1)}, & \text{otherwise.} \end{cases} \quad (8)$$

The time context factor $F_{time}(p, t) = \exp(A_{p,k(t)})$ favors those active POIs, where $k(t)$ is the corresponding time slot of time stamp t.

The selection of time slot duration involves a tradeoff between accuracy of POI activeness and sparsity of map search queries. Time slots of shorter duration, e.g., in minutes, can record finer-grained POI activeness, but require more map search query data. In this study, we consider two-hour time slots, i.e., $T = 12$. Note that two hours is a reasonable length of time for most POI visit purposes.

Finally, the *popularity* context factor $F_{popu}(p)$ captures people's visiting preference for popular POIs. It is also evaluated from map search query data as popular POIs usually have more map queries: $F_{popu}(p) = \log\left(\sum_k Q_{p,k} + 1\right)$.

Once the distance, time and popularity factors are evaluated for all stays and their candidate POIs (the top-100 nearest POIs around the stay location), we perform a Z-score normalization on each of the three factors. This is to alleviate the possible influence of different scales of the domain knowledge context factors.

3.4 Training and Inference with Context Fusion

We finally present the training and inference procedures with context fusion. For each candidate POI p of a stay of user u at location l and time stamp t, NCF evaluates the context factors as stated in above and concatenates these factors into a factor vector $\mathbf{f}$ in a pre-defined order:

$$\begin{aligned} \mathbf{f} = [&F^{(1)}_{pref}(u,p), F^{(1)}_{pbtr}(\mathcal{R}^t_u, p), F^{(1)}_{ubtr}(\mathcal{R}^t_u, u), \\ &\dots, \\ &F^{(H)}_{pref}(u,p), F^{(H)}_{pbtr}(\mathcal{R}^t_u, p), F^{(H)}_{ubtr}(\mathcal{R}^t_u, u), \\ &F_{dist}(p,l), F_{time}(p,t), F_{popu}(p)]. \end{aligned} \quad (9)$$

The factor vector is then fed into a feed-forward fusion network to derive the un-normalized visit probability. We adopt an MLP with three layers of adaptive weights and applying the ReLU nonlinearity for context fusion:

$$Pr(u,l,t,p) = \mathbf{W}_3 \mathsf{ReLU}(\mathbf{W}_2 \mathsf{ReLU}(\mathbf{W}_1 \mathbf{f} + \mathbf{b}_1) + \mathbf{b}_2), \quad (10)$$

where $\mathbf{W}_1 \in \mathbb{R}^{d \times (3H+3)}$, $\mathbf{W}_2 \in \mathbb{R}^{d \times d}$, $\mathbf{W}_3 \in \mathbb{R}^{1 \times d}$ and $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{R}^d$ are the learnable parameters.

We train our NCF with all stays whose ground-truth visited POIs are included in the top-100 most nearest candidate POIs. For each training stay, let $\hat{\mathbf{y}}$ be the vector consisting of the visit probabilities of all candidate POIs and $\mathbf{y}$ be the corresponding one-hot vector of the index of the ground-truth POI. We then minimize the cross-entropy loss between $\mathbf{y}$ and softmax($\hat{\mathbf{y}}$). For inference, we compute the visit probabilities with our NCF for all candidate POIs. Afterward, a stay is assigned the POI p whose visit probability is the highest among all candidates.

Remarks. In this work, we choose a supervised setting for raw mobility annotation. The advantages are as follows. First, we can better exploit the the preference context between users and POIs, which plays a very important role in raw mobility annotation. Second, we can develop an expressive neural model and train it in an end-to-end manner. As a result, our NCF is very suitable for dealing with annotation scenarios when mobility records are generated within areas with extremely dense POI layout. In practice, there exist a number of ways to collect labels to ensure supervised learning. For instance, we can collect users' check-ins to POIs as labels or we can exploit an easy-first strategy such that we derive some seed labels by annotating the easy stays, e.g., those having very few candidate POIs.

Table 2
Data set statistics

Description	# of stays	# of users	# of POIs	# of regions
BEIJING	436,728	26,917	1,341,663	62,534
NYC	146,325	1,083	318,162	45,935

4 EXPERIMENTS

In this section, we present an experimental study of our NCF approach. Using two real-life data sets, we conduct six sets of experiments to evaluate: (a) the overall effectiveness for raw mobility annotation, (b) the effectiveness of the different components in NCF, (c) the efficiency, (d) the parameter sensitivity, (e) the utility of the obtained POI-based human mobility, and (f) the effectiveness for unseen users.

4.1 Experimental Setups

We first introduce the settings of the experimental study.

Data sets. We chose two data sets to test our model.

(1) BEIJING was obtained from a third-party map service platform and was produced based on the POI data, road-segmented region data, and anonymous map search query and raw mobility records during July, 2018 in Beijing. Stays were identified from raw mobility records with ξ_D, ξ_T and ξ_S set to 200 meters, 2 hours and 10 minutes, respectively (Section 2). Ground-truth visited POIs were determined (i) with an empirical rule combining both mobility and map search query data and (ii) by human experts. We finally removed stays without annotated POIs.

(2) NYC was produced based on a public Foursquare check-in data set collected from April 12, 2012 to February 16, 2013 [22]. Since the check-in locations were very close to the visited POIs, following [10], we injected noises drawn from a uniform distribution ($\mu = 0, \delta = 0.0002$) to the check-in locations. We also collected the basic POI information and the numbers of likes to POIs with Foursquare developers APIs.[1] Note that the latter was used to estimate the popularity context factor. Due to the quota limitations of APIs, we were unable to obtain the POI activeness data. Hence, we did not use the time factor on NYC. Regions were derived based on the road network downloaded from NYC Open Data.[2] We finally filtered out the check-ins not within any regions and treated each remaining as a stay. We will release our NYC data in the future.

[1] https://developer.foursquare.com/

[2] https://data.cityofnewyork.us/City-Government/NYC-Street-Centerline-CSCL-/exjm-f27b

Table 3
Accuracy (Acc) comparison with different fraction f of training data

Data set	Method	10%	20%	30%	40%	50%	60%	70%	80%	90%	Avg.
BEIJING	Dist	0.0678	0.0678	0.0678	0.0678	0.0681	0.0675	0.0675	0.0676	0.0673	0.0677
	HMM	0.1287	0.1603	0.1820	0.1967	0.2094	0.2187	0.2270	0.2354	0.2434	0.2002
	LTR	0.2760	0.3294	0.3554	0.3675	0.3790	0.3899	0.3934	0.3961	0.4065	0.3659
	MRF	0.2142	0.2819	0.3245	0.3571	0.3804	0.3985	0.4149	0.4286	0.4414	0.3602
	GE	0.2212	0.3072	0.3411	0.3583	0.3662	0.3720	0.3770	0.3797	0.3817	0.3450
	NCF(3)	**0.3853**	**0.4229**	**0.4491**	**0.4829**	**0.4984**	**0.5053**	**0.5284**	**0.5343**	**0.5452**	**0.4835**
NYC	Dist	0.2683	0.2681	0.2678	0.2687	0.2683	0.2680	0.2671	0.2709	0.2666	0.2682
	HMM	0.2811	0.2966	0.3132	0.3307	0.3465	0.3589	0.3708	0.3823	0.3914	0.3413
	LTR	0.3939	0.4194	0.4252	0.4484	0.4580	0.4522	0.4712	0.4764	0.4896	0.4483
	MRF	0.2984	0.3681	0.4188	0.4620	0.4927	0.5230	0.5429	0.5652	0.5810	0.4725
	GE	0.3677	0.3387	0.3394	0.3377	0.3421	0.3414	0.3368	0.3414	0.3398	0.3428
	NCF(3)	**0.6120**	**0.6728**	**0.7192**	**0.7520**	**0.7664**	**0.7797**	**0.7981**	**0.8061**	**0.8133**	**0.7466**

The standard deviations of all reported Acc are less than 0.03, and NCF(3) significantly outperforms other baselines at the 0.01 level, paired t-test.

For both data sets, we randomly split f of stays for training and used the rest stays for testing, i.e., stay-level split. If validation was required, 10% of the training data was left out and used for validation. Table 2 lists the statistics of our BEIJING and NYC data. It is noteworthy that both our data sets are essentially based on map and trip services, and the results of our experimental study should be better interpreted under this specific situation.

Metric. We adopted the accuracy (Acc) metric to evaluate the effectiveness, which is the ratio of the number N_{ca} of correctly annotated test stays to the number N_{all} of all test stays: Acc $= N_{ca}/N_{all}$.

Algorithms. We compared our NCF approach with the following baselines and variants of NCF.

- Dist utilizes spatial information only and annotates each stay to the nearest POI around the stay location.
- HMM trains a hidden Markov model to learn transition relationships between POI categories [11]. After deriving the most likely category, it then annotates a stay to the nearest POI of that category.
- LTR is a learning to rank model, which trains a LambdaMART model to rank POIs near a location [9]. It then uses the top-1 ranked POIs for annotation. We re-implemented six of the nine features originally developed in [9], excluding Creator, Mayor and Friends-Here-Now since they are not available in our data.
- MRF constructs a Markov random field and annotates each stay with a POI via minimizing the energy [10]. It captures personal preferences by enforcing consistency in spatially- or temporally-close stays. For fairness, we used the supervised version provided by the authors.
- GE is a classic non-sequential approach to location recommendation [17]. It constructs four graphs and learns POI, region and time embeddings via preserving graph structures. It finally evaluates the inner product of the corresponding embeddings for annotation.
- DKF is a simplified version of NCF, which only fuses the domain knowledge factors with the fusion network.
- NCF(H) the H-head version of NCF.

Implementations. We implemented the variants of NCF in Python and used the Adam optimizer with a batch size of 256. The learning rate γ increased in the first $warmup$ steps and then decreased [27], i.e., $\gamma = 3 \cdot d^{-0.5} \cdot \min\{step \cdot warmup^{-1.5}, step^{-0.5}\}$, where $step$ denoted the step number and $warmup$ was fixed to 1000. We employed three types of regularization to prevent overfitting: (a) an L_2 regularization with $\lambda = 5 \times 10^{-4}$ on all trainable embedding and model parameters, (b) a dropout after each dense layer of MLPs (except for the last layer of the fusion network), and (c) an early stopping if the Acc on validation set did not increase in successive 5 epochs. The dropout probability P_{drop} was chosen based on the fraction f of training stays, i.e., a high $P_{drop} = 0.6$ for $f \leq 30\%$, a low $P_{drop} = 0.2$ for $f \geq 70\%$, and a median $P_{drop} = 0.4$ otherwise. Parameter ϕ was fixed to a moderate 0.005. Finally, we set the number H of heads, the number N_r of regions in $\mathcal{R}_u^t$ and the number d of dimensions to 3, 10 and 64 by default, respectively. We will test the parameter sensitivity in our experiments.

The LTR approach was implemented with the Java RankLib[3] library. We trained 2,000 trees, used the Z-score normalization on features, applied an early stop after 20 rounds without performance gain on validation set, and fixed the shrinkage parameter to 0.3 if $f \leq 60\%$ and 0.2 otherwise. The rest algorithms were implemented in C++ following their recommended settings.

All experiments were conducted on a workstation with Intel Xeon 2.0GHz CPUs, 200 GB of main memory and Tesla P40 GPUs. When quantity measures were evaluated, the test was repeated over 5 times using different train-test splits and the average result was reported.

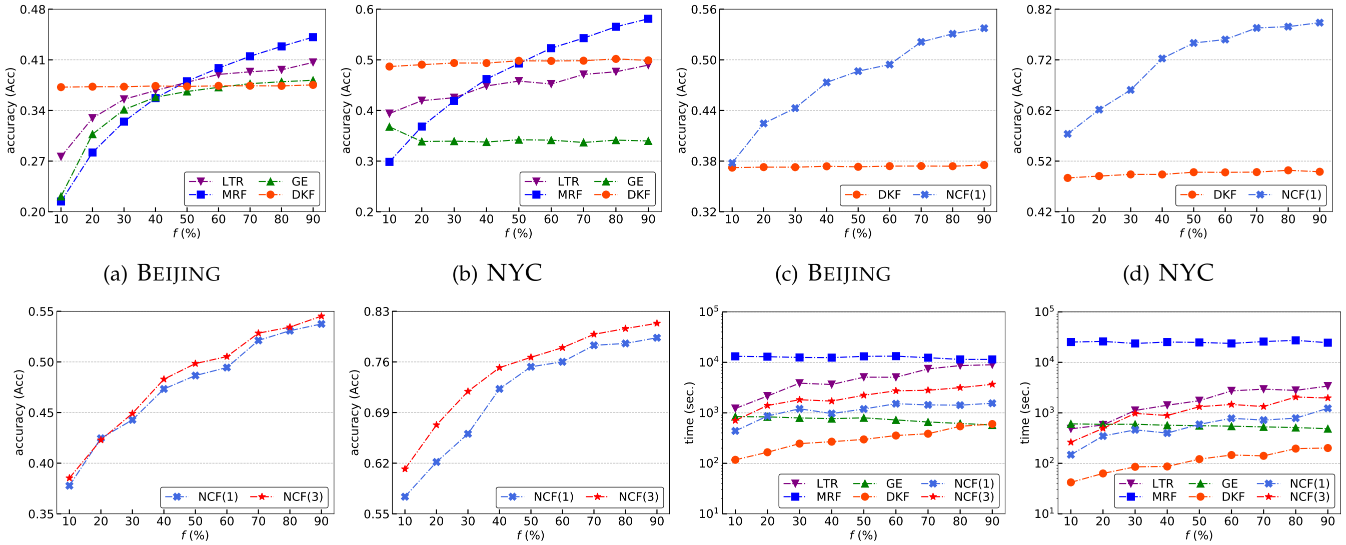
4.2 Experimental Results

We next present our experimental results.

Exp-1: Effectiveness comparison. In the first set of experiments, we evaluate the overall effectiveness of NCF and the five baselines for raw mobility annotation. We considered the multi-head version of our approach, i.e., NCF(3). To ensure a comprehensive comparison, we varied the fraction f of training stays from 10% to 90%. The resulting Acc of all tested approaches are reported in Table 3. Note that each Acc is the average result of five tests.

Overall, considering more visiting context information can generally promote the effectiveness of raw mobility annotation. Observe that the transition relationships between POI categories has already substantially improved the Acc of HMM compared with the spatial-only Dist approach. Moreover, the more complex contexts considered by LTR, MRF, and GE make the three approaches further better than both Dist and HMM.

[3] https://sourceforge.net/p/lemur/wiki/RankLib/



(a) BEIJING (b) NYC (c) BEIJING (d) NYC

(e) BEIJING (f) NYC (g) BEIJING (h) NYC

Figure 5. Ablation study (a)–(f) and efficiency evaluation (g)–(h)

The average effectiveness of LTR and MRF are comparable on BEIJING and NYC. However, their performance gap varies with f: LTR works better than MRF given limited training data, e.g., $f \leq 30\%$ while MRF outperforms LTR when $f \geq 50\%$. This is because LTR trains a unified model for all users together while MRF learns to capture the distinct POI-visiting pattern of each individual. The personalized pattern learning mechanism of MRF becomes more effective with the increment of f.

Raw mobility annotation bears some similarity to location recommendation and the location recommendation approach GE can also annotate mobility records after some minor revision. However, GE only considers the complex co-occurrence-based contexts between POI, region and time while ignores other basic contexts such as distance and POI popularity. As a result, GE has an inherent limitation, especially on NYC, compared with other annotation methods such as LTR and MRF.

Finally, our NCF approach which fuses various contexts significantly outperforms all baselines at the 0.01 p-value level (paired t-test) on both BEIJING and NYC. Indeed the Acc of NCF(3) is on average (614%, 142%, 32%, 34%, 40%) and (178%, 119%, 67%, 58%, 118%) higher than (Dist, HMM, LTR, MRF, GE) on BEIJING and NYC, respectively. Such improvement demonstrates that context fusion is an effective tool for raw mobility annotation.

Exp-2: Ablation study. In the second set of experiments, we present an ablation study to evaluate the effectiveness of the domain knowledge factors, the representation learning factors and the multi-head architecture of NCF. Again, the fraction f was varied from 10% to 90%.

Exp-2.1: Domain knowledge factors. To evaluate the effectiveness of domain knowledge factors, we tested and compared the Acc of DKF with LTR, MRF and GE. The results are reported in Figs. 5(a) & 5(b). We omitted Dist and HMM due to their low Acc reported in Table 3.

When varying f, the Acc of LTR, MRF and GE increase with the increment of f, except for GE on NYC. Note that GE simply uses the visited POIs to represent users, which might be inaccurate given the long spanning time of NYC. Differently, DKF can learn to effectively fuse the domain knowledge factors for raw mobility annotation given very limited data. This property makes DKF better than other tested methods when $f \leq 40\%$ on both data sets. In summary, the domain knowledge factors can successfully identify some general patterns for raw mobility annotation.

Exp-2.2: Representation learning factors. To evaluate the effectiveness of representation learning factors, we tested and compared the Acc of DKF and NCF(1). The results are reported in Figs. 5(c) & 5(d).

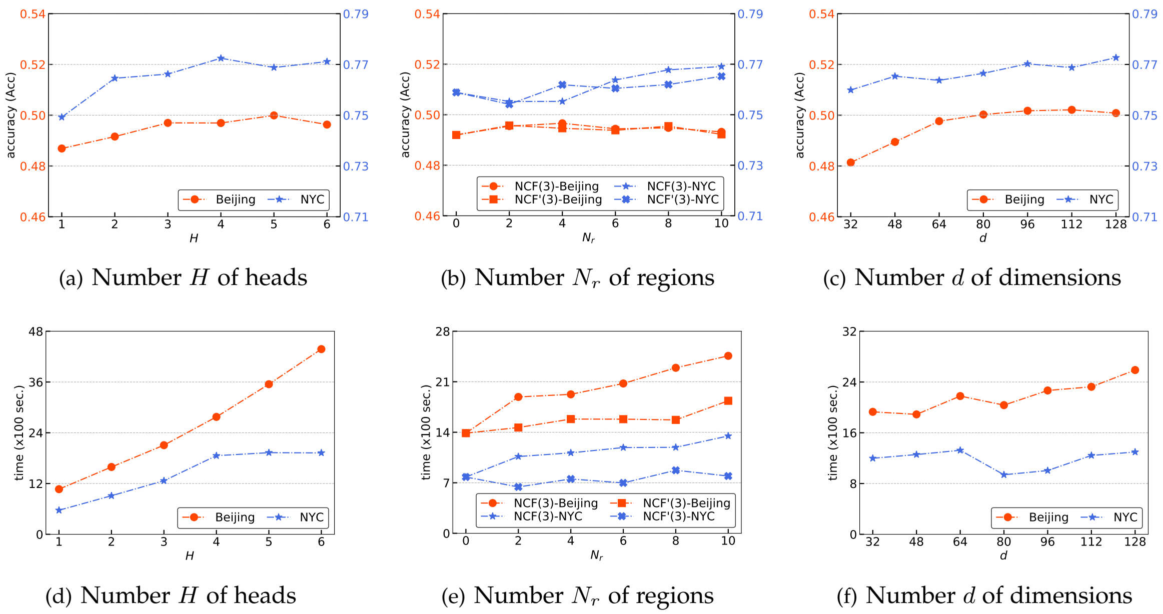
When varying f, the Acc of NCF(1) is consistently higher than DKF on both data sets. The gap of Acc also increases with the increment of f. Overall, compared with DKF, our representation learning factors increase the Acc by 27.5% and 44.7% on average on BEIJING and NYC, respectively.

Exp-2.3: Multi-head architecture. To evaluate the effectiveness of the multi-head architecture, we tested and compared the Acc of NCF(1) and NCF(3) reported in Figs. 5(e) & 5(f).

When varying f, the multi-head architecture promotes the effectiveness of our NCF approach in almost all cases in our tests, except for $f = 20\%$ on BEIJING. The improvement is more significant on NYC (4.12% on average) than BEIJING (1.47% on average). Recall that the NYC data spans for a longer time than BEIJING and the POI-visiting patterns on NYC are very likely to be more complex accordingly. As a consequence, the multi-head architecture introduces more performance gain on NYC.

Exp-3: Efficiency comparison. In the third set of tests, we evaluate the overall efficiency of NCF. We tested the running time of DKF, NCF(1) and NCF(3) as well as baselines LTR, MRF and GE, with f varied from 10% to 90%. Similar to Exp-2.1, Dist and HMM are omitted due to their effectiveness. The results are reported in Figs. 5(g) & 5(h).

When varying f, the running time of GE decreases while the ones of other approaches increase with the increment of f. Note that the number of samples trained by GE is fixed



(a) Number H of heads (b) Number N_r of regions (c) Number d of dimensions

(d) Number H of heads (e) Number N_r of regions (f) Number d of dimensions

Figure 6. Parameter sensitivity (the left and right y-axes of Figs. 6(a)–6(c) correspond to the Acc on BEIJING and NYC, respectively)

and it costs less time on inference with larger f. Except for GE, DKF runs the fastest, followed by NCF(1), NCF(3), LTR and MRF, respectively. Our compete NCF(3) is still faster than the rest two mobility annotation approaches LTR and MRF. Indeed, the running time of (DKF, GE, NCF(1), LTR, MRF) is on average (0.15, 0.33, 0.52, 2.28, 5.57) and (0.10, 0.46, 0.51, 1.60, 21.03) times of the running time of NCF(3) on BEIJING and NYC, respectively.

Exp-4: Parameter sensitivity. In the fourth set of experiments, we evaluate the parameter sensitivity of NCF. Fixing f to 50%, we tested the impacts of the number H of heads, the number N_r of regions and the number d of dimensions. In each experiment, we varied the tested parameter while fixed others to their default values.

Exp-4.1: Impacts of H. To evaluate the impacts of the number H of heads, we varied H from 1 to 6 and tested the Acc and running time reported in Figs. 6(a) & 6(d).

When varying H, the Acc of NCF(H) first increases with the increment of H when $H \leq 4$, due to the enlarged model expressiveness. Further increasing H, the Acc might slightly decrease. In these cases, our NCF(H) tends to be overfit with superabundant parameters. It turns out that a moderate H, e.g., 3 or 4, is reliable for enhancing the expressive power. With larger H, the model has a potential to learn more, but there is also a risk of overfitting.

When varying H, the running time increases with the increment of H reasonably. Indeed, the running time with $H = (2, 3, 4, 5, 6)$ is on average (1.5, 2.0, 2.6, 3.3, 4.1) and (1.6, 2.2, 3.3, 3.4, 3.4) times of the one of $H = 1$ on BEIJING and NYC, respectively.

Exp-4.2: Impacts of N_r. To evaluate the impacts of the number N_r of regions in $\mathcal{R}_u^t$, we varied N_r from 0 to 10 and tested the Acc and running time reported in Figs. 6(b) & 6(e). Note the for $N_r = 0$ we removed the two transition factors and we also considered another variant NCF′ which used $N_r/2$ previous and $N_r/2$ subsequent regions as $\mathcal{R}_u^t$.

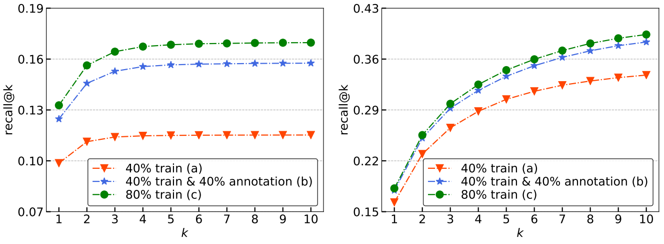
Incorporating transition context factors via regions has a positive impact on the effectiveness of NCF. However, the best improvement is obtained with different N_r on BEIJING and NYC. For BEIJING, a relatively small N_r is preferred while considering more regions is better for NYC. Recall that BEIJING records dense raw human mobility and a small number of previously stayed regions can provide the most meaningful transitional clues. On the other hand, the NYC data is obtained from the sparse check-in data and collecting clues from more regions is generally more effective. Finally, the effectiveness of NCF and NCF′ are close in general.

The running time of NCF increases with the increment of N_r reasonably. Since NCF′ requires less epoches for training, it runs (1.3, 1.5) times faster than NCF on (BEIJING, NYC). Overall, NCF is more general, e.g., annotating new mobility records, at the expense of more training computation.

Exp-4.3: Impacts of d. To evaluate the impacts of the number d of dimensions, we varied d from 32 to 128 and tested the Acc and running time reported in Figs. 6(c) & 6(f).

When varying d, the Acc generally increases with the increment of d. From our tests, we recommend to choose d within $[64, 128]$ for NCF. The running time also generally increases with the increment of d, except for the drop at $d = 80$. It turns out that NCF with $d \geq 80$ can stop after less epoches compared with $d < 80$. As a result, the running time with different d is overall comparable: the relative proportion in Fig. 6(f) is bounded by 1.3.

Exp-5: Utility of POI-based human mobility. In the fifth set of experiments, we illustrate the utility of the obtained POI-based human mobility with a POI recommendation example. To avoid introducing model biases, we developed a simple recommendation strategy. Specifically, given a user u and a time t of a day (e.g., 5:00 P.M.), we retrieved the historical POI visit records $\mathcal{V}_u^t = \{(p_i, t_i)\}$ of user u such that the time t_i of a day was temporally-close to t (within 1 hour). We then recommended the POIs in $\mathcal{V}_u^t$ according to the sum of temporal closeness $C(p) = \sum_{(p_i,t_i)\in\mathcal{V}_u^t, p_i=p} \exp(\Delta(t, t_i)/3600)$, where $\Delta(t, t_i)$



(a) BEIJING (b) NYC

Figure 7. Utility of raw mobility annotation

was the time span between t and t_i in second.

We used the latest 20% of POI visit records of each user for testing. For the remaining 80% of visit records of each user, we randomly selected 40% as observed POI visits and the rest 40% were regarded as unannotated stays. We obtained three recommendation models: (a) using the observed POI visits only (b) using both observed POI visits and the annotation results by NCF(3) on unannotated stays, and (c) using both observed POI visits and the ground-truth POIs associated with unannotated stays. Note that the last represented the best performance that could be achieved by our recommendation strategy. We adopted Recall@k with $1 \leq k \leq 10$ to evaluate the recommendation performance. The results are reported in Fig. 7.

On both BEIJING and NYC, the obtained POI-based human mobility consistently improves the effectiveness of POI recommendation for all tested k. Indeed, the Recall@k is increased by 34.6% and 11.2% on BEIJING and NYC on average, respectively. Furthermore, with the exploited strategy, the performance based on the POI-based human mobility by NCF already approaches the best, which is 93.6% and 97.7% of the best on BEIJING and NYC on average, respectively.

Exp-6: Effectiveness comparison with user-level split. In the last set of experiments, we evaluate the annotation effectiveness of all approaches with user-level split. Note that it is usually more difficult to deal with the new setting compared with stay-level since we have no direct clues to model preference and transition patterns of users in test data. We trained models with data from half of users and annotated the stays of the rest users, i.e., $f = 50\%$. Note that HMM learns transition relationships between POI categories from the training users and applies the knowledge to the testing users. Besides, MRF degenerates to the unsupervised version as no labels are available for testing users. The results are reported in Table 4.

On both data sets, the Acc of all approaches decreases with user-level split, except for Dist. The reason of degradation for HMM is that HMM does not have a precise starting point to maximize the posterior probability of a Markov chain. While for LTR, this is because a useful feature, i.e., the number of previous visits from the user at the POI, becomes meaningless. On the other hand, both MRF and GE perform poorly since errors propagate through the spatial and temporal constraints of MRF and GE models a user with the visited POIs. Finally, owing to our context fusion strategy, NCF(3) is at least (26%, 95%) better than all competitors on (BEIJING, NYC) in the new setting.

Summary. From our tests, we find the following.

Table 4
Accuracy (Acc) comparison with Stay- and User-level splits ($f = 50\%$)

Data set	Dist	HMM	LTR	MRF	GE	NCF(3)
BEIJING(S)	0.0681	0.2094	0.3790	0.3804	0.3662	**0.4984**
BEIJING(U)	0.0679	0.1486	0.2774	0.0685	0.1046	**0.3495**
NYC(S)	0.2683	0.3465	0.4580	0.4927	0.3421	**0.7664**
NYC(U)	0.2669	0.2787	0.2361	0.0988	0.0686	**0.5440**

NCF(3) significantly outperforms other baselines at the 0.01 level, paired t-test, with both stay- and user-level splits.

(1) Our NCF approach which fuses various context factors consistently outperforms other raw mobility annotation approaches. The Acc of our complete NCF(3) is on average (614%, 142%, 32%, 34%, 40%) and (178%, 119%, 67%, 58%, 118%) higher than (Dist, HMM, LTR, MRF, GE) on BEIJING and NYC, respectively.
(2) Both the domain knowledge factors and the representation learning factors are effective for raw mobility annotation. Besides, the adopted multi-head architecture also improves the accuracy of annotation.
(3) Our NCF is also efficient for raw mobility annotation. It runs faster than LTR and MRF. Compared with GE, the extra time used by NCF is affordable for achieving the much better effectiveness.
(4) We verify the utility of the obtained POI-based human mobility in a POI recommendation example.

5 RELATED WORK

Location-based human mobility. From a purely spatiotemporal perspective, the location-based human mobility focuses on movements from one location to another. Basic rules that govern our daily movements are identified [3], [1], as well as the individual or group mobility patterns regarding the locations people tend to visit sequentially [2] or periodically [32]. At aggregated level, Gonzalez et al. find that human trajectories show a high degree of temporal and spatial regularity, i.e., humans follow simple reproducible mobility patterns [3]. Along this, Song et al. find a 93% potential predictability in user mobility, by measuring the entropy of trajectories [4]. On the other hand, McInerney et al. analyze an individual's mobility patterns and identify temporary departures from routine [33]. Oliveira et al. uncover people's tendency to revisit few favorite venues using the shortest-path available [34]. For location-level mobility prediction, Feng et al. propose an attentional recurrent network [35] and Baumann et al. study selecting individual and population models [36]. Besides, Li et al. consider location inference on social media[37].

Activity-based human mobility. Studies in this category essentially try to explain the reasons behind people's moves [5], [6], [7], [8]. From a computational point of view, Alvares et al. identify the needs of enriching trajectories with semantics to simplify queries, analysis, and mining of moving objects [5]. They then propose a data pre-processing model that associates sample points in trajectories with geographic data points. Jiang et al. extract activity-based human mobility patterns, e.g., *Home–Work–Home*, from mobile phone call detail record data [6]. These patterns can assist transportation and planning agencies to understand the human activity patterns in cities. Geo-tagged social media are another source to mine activity-based mobility [7], [8]. Specifically, Zhang et al. obtain mobility models by alternating between user grouping and mobility modeling [7]. Zhu et al. learn a multi-modal spherical hidden Markov model for semantics-rich human mobility [8].

POI-based human mobility. Mobility is also analyzed together with POIs. Annotation is a typical example, e.g., [11], [9], [10] and ours. The existing annotation methods are mainly designed for sparse mobility and usually exploit limited types of contexts. For instance, Shaw et al. formalize the annotation problem in a learning to rank framework, considering basic POI attributes and users' POI-visit histories [9]. Historical POI visits are further exploited in the Markov random field for annotation [10]. It captures personal preferences by enforcing consistency in spatially- or temporally-close mobility records. While [11] is designed for the dense raw mobility, it only learns the transition relationships between POI categories with a hidden Markov model. Domain knowledge contexts are effective for annotation and have already been explored. The distance context is exploited in [11], [10] and all the three contexts are used by [9] as learning to rank features.

Our work is different from these studies in two aspects. First, we fuse various context factors in people's dense POI-visiting behaviors. Second, we adopt a neural network which captures the hidden preference and transition structures via representation learning and fuses context factors in a feed-forward network.

POI recommendation. Another line of related work is POI recommendation. There have been extensive studies for the task, exploiting different strategies like temporal effects [20], graph embedding [17], sequential modeling [38] and preference context modeling [19], to name a few. To some extent, human mobility annotation and POI recommendation are similar, in the sense that we can recommend the nearby POIs as annotation. Indeed, some approaches to location recommendation can be applied for annotation after necessary adaptation. Our work also differs from those for POI recommendation such that we know the stay location and time information in advance and we delicately recognize and model the various contexts of a visit from different perspectives, e.g., preference, transition, distance, time, and popularity, for annotation.

6 Concluding Remarks

In this paper we studied raw mobility annotation to obtain high-quality POI-based human mobility. The dense trajectories as well as the semantics embedded in POIs can better support many mobile analytic tasks. We proposed the first neural model which fused various key context factors in people's POI-visiting behaviors for the task. These factors were either derived via representation learning or inspired by domain knowledge. Notably, we utilized an attention mechanism to deal with the randomized effect in transitions of raw human mobility and adopted a multi-head architecture to enhance model expressiveness. Finally, our experimental study on two real-life data sets demonstrated the effectiveness and efficiency of our approach as well as the utility of the obtained POI-based human mobility.

Acknowledgments

This work is supported in part by NSFC 71531001 & 61925203 & U1636210 & 61421003, Beijing Advanced Innovation Center for Big Data and Brain Computing. For any correspondence, please refer to Shuai Ma and Hui Xiong.

References

[1] H. Barbosa, M. Barthelemy, G. Ghoshal, C. R. James, M. Lenormand, T. Louail, R. Menezes, J. J. Ramasco, F. Simini, and M. Tomasini, "Human mobility: Models and applications," *Physics Reports*, vol. 734, pp. 1 – 74, 2018.

[2] F. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi, "Trajectory pattern mining," in *SIGKDD*, 2007.

[3] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi, "Understanding individual human mobility patterns," *Nature*, vol. 453, no. 7196, p. 779, 2008.

[4] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási, "Limits of predictability in human mobility," *Science*, vol. 327, no. 5968, pp. 1018–1021, 2010.

[5] L. O. Alvares, V. Bogorny, B. Kuijpers, J. A. F. de Macêdo, B. Moelans, and A. A. Vaisman, "A model for enriching trajectories with semantic geographical information," in *GIS*, 2007.

[6] S. Jiang, J. F. Jr., and M. C. González, "Activity-based human mobility patterns inferred from mobile phone data: A case study of singapore," *IEEE Trans. Big Data*, vol. 3, no. 2, pp. 208–219, 2017.

[7] C. Zhang, K. Zhang, Q. Yuan, L. Zhang, T. Hanratty, and J. Han, "Gmove: Group-level mobility modeling using geo-tagged social media," in *SIGKDD*, 2016.

[8] W. Zhu, C. Zhang, S. Yao, X. Gao, and J. Han, "A spherical hidden markov model for semantics-rich human mobility modeling," in *AAAI*, 2018.

[9] B. Shaw, J. Shea, S. Sinha, and A. Hogue, "Learning to rank for spatiotemporal search," in *WSDM*, 2013.

[10] F. Wu and Z. Li, "Where did you go: Personalized annotation of mobility records," in *CIKM*, 2016.

[11] Z. Yan, D. Chakraborty, C. Parent, S. Spaccapietra, and K. Aberer, "Semantic trajectories: Mobility data computation and annotation," *ACM Trans. Intell. Syst. Technol.*, vol. 4, no. 3, pp. 49:1–49:38, 2013.

[12] E. Thuillier, L. Moalic, S. Lamrous, and A. Caminada, "Clustering weekly patterns of human mobility through mobile phone data," *IEEE Transactions on Mobile Computing*, vol. 17, no. 4, pp. 817–830, 2018.

[13] Y. Zheng, Y. Liu, J. Yuan, and X. Xie, "Urban computing with taxicabs," in *UbiComp*, 2011.

[14] S. Eubank, H. Guclu, V. A. Kumar, M. V. Marathe, A. Srinivasan, Z. Toroczkai, and N. Wang, "Modelling disease outbreaks in realistic urban social networks," *Nature*, vol. 429, no. 6988, p. 180, 2004.

[15] J. Liu, L. Sun, W. Chen, and H. Xiong, "Rebalancing bike sharing systems: A multi-source data smart optimization," in *SIGKDD*, 2016.

[16] D. Chen, C. S. Ong, and L. Xie, "Learning points and routes to recommend trajectories," in *CIKM*, 2016.

[17] M. Xie, H. Yin, H. Wang, F. Xu, W. Chen, and S. Wang, "Learning graph-based poi embedding for location-based recommendation," in *CIKM*, 2016.

[18] Y. Liu, C. Liu, X. Lu, M. Teng, H. Zhu, and H. Xiong, "Point-of-interest demand modeling with human mobility patterns," in *SIGKDD*, 2017.

[19] T.-N. Doan and E.-P. Lim, "Pacela: A neural framework for user visitation in location-based social networks," in *the 26th Conference on User Modeling, Adaptation and Personalization*, 2018.

[20] H. Gao, J. Tang, X. Hu, and H. Liu, "Exploring temporal effects for location recommendation on location-based social networks," in *RecSys*, 2013.

[21] J.-B. Griesner, T. Abdessalem, and H. Naacke, "Poi recommendation: Towards fused matrix factorization with geographical and temporal influences," in *RecSys*, 2015.

[22] D. Yang, D. Zhang, V. W. Zheng, and Z. Yu, "Modeling user activity preference by leveraging user spatial temporal characteristics in lbsns," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 1, pp. 129–142, 2015.

[23] S. Migliorini, M. Gambini, and A. Belussi, "A blockchain-based solution to fake check-ins in location-based social networks," in *the 3rd ACM SIGSPATIAL International Workshop on Analytics for Local Events and News*, 2019.

[24] H. Liu, T. Li, R. Hu, Y. Fu, J. Gu, and H. Xiong, "Joint representation learning for multi-modal transportation recommendation," in *AAAI*, 2019.

[25] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *ICLR*, 2014.

[26] Q. Li, Y. Zheng, X. Xie, Y. Chen, W. Liu, and W.-Y. Ma, "Mining user similarity based on location history," in *SIGSPATIAL GIS*, 2008.

[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017.

[28] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[29] N. J. Yuan, Y. Zheng, X. Xie, Y. Wang, K. Zheng, and H. Xiong, "Discovering urban functional zones using latent activity trajectories," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 3, pp. 712–725, 2015.

[30] "Radial basis function kernel." [Online]. Available: https://en.wikipedia.org/wiki/Radial_basis_function_kernel

[31] M. Xu, T. Wang, Z. Wu, J. Zhou, J. Li, and H. Wu, "Demand driven store site selection via multiple spatial-temporal data," in *SIGSPATIAL GIS*, 2016.

[32] Q. Yuan, W. Zhang, C. Zhang, X. Geng, G. Cong, and J. Han, "PRED: periodic region detection for mobility modeling of social media users," in *WSDM*, 2017.

[33] J. McInerney, S. Stein, A. Rogers, and N. R. Jennings, "Breaking the habit: Measuring and predicting departures from routine in individual human mobility," *Pervasive and Mobile Computing*, vol. 9, no. 6, pp. 808 – 822, 2013.

[34] E. M. R. Oliveira, A. C. Viana, C. Sarraute, J. Brea, and I. Alvarez-Hamelin, "On the regularity of human mobility," *Pervasive and Mobile Computing*, vol. 33, pp. 73 – 90, 2016.

[35] J. Feng, Y. Li, C. Zhang, F. Sun, F. Meng, A. Guo, and D. Jin, "Deepmove: Predicting human mobility with attentional recurrent networks," in *WWW*, 2018.

[36] P. Baumann, C. Koehler, A. K. Dey, and S. Santini, "Selecting individual and population models for predicting human mobility," *IEEE Transactions on Mobile Computing*, vol. 17, no. 10, pp. 2408–2422, 2018.

[37] P. Li, H. Lu, N. Kanhabua, S. Zhao, and G. Pan, "Location inference for non-geotagged tweets in user timelines," *IEEE Transactions on Knowledge and Data Engineering*, 2018.

[38] Q. Liu, S. Wu, L. Wang, and T. Tan, "Predicting the next location: A recurrent model with spatial and temporal contexts," in *AAAI*, 2016.

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

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

Xinjiang Lu is currently a researcher at Business Intelligent Lab, Baidu Inc., Beijing, China. He received his Ph.D. degree in Computer Science from Northwestern Polytechnical University in 2018, M.S. degree in Software Engineering from Northwestern Polytechnical University in 2011 and B.E. degree from Xinjiang University in 2007. His recent research interests include data mining and mobile intelligence.

Hengzhu Zhu (SM'19) is currently a senior data scientist at Baidu Inc. He received the Ph.D. degree in 2014 and B.E. degree in 2009, both in Computer Science from University of Science and Technology of China (USTC), China. His general area of research is data mining and machine learning, with a focus on developing advanced data analysis techniques for emerging applied business research. He has published prolifically in refereed journals and conference proceedings, including IEEE Transactions on Knowledge and Data Engineering (TKDE), IEEE Transactions on Mobile Computing (TMC), ACM Transactions on Knowledge Discovery from Data (TKDD), ACM SIGKDD, ACM SIGIR, WWW, IJCAI, and AAAI. He has served regularly on the organization and program committees of numerous conferences, including as a program Co-Chair of the KDD Cup-2019 Regular ML Track, and a founding co-chair of the first International Workshop on Organizational Behavior and Talent Analytics (OBTA-2018) and the International Workshop on Talent and Management Computing (TMC-2019), which were held in conjunction with ACM SIGKDD-2018 and ACM SIGKDD-2019 respectively. He was the recipient of the Best Student Paper Award of KSEM-2011, WAIM-2013, CCDM-2014, and the Best Paper Nomination of ICDM-2014. He is the senior member of IEEE, ACM, and CCF, and the committee member of CCF Task Force on Big Data.

Shuai Ma is a professor at the School of Computer Science and Engineering, Beihang University, China. He obtained his PhD degrees from University of Edinburgh in 2010, and from Peking University in 2004, respectively. He was a post-doctoral research fellow in the database group, University of Edinburgh, a summer intern at Bell labs, Murray Hill, USA and a visiting researcher of MSRA. He is a recipient of the best paper award for VLDB 2010 and the best challenge paper award for WISE 2013. He is an Associate Editor of VLDB Journal since 2017. His current research interests include database theory and systems, and big data.

Hui Xiong (F'19) is currently a Full Professor at the Rutgers, the State University of New Jersey, where he received the 2018 Ram Charan Management Practice Award as the Grand Prix winner from the Harvard Business Review, RBS Dean's Research Professorship (2016), the Rutgers University Board of Trustees Research Fellowship for Scholarly Excellence (2009), the ICDM Best Research Paper Award (2011), and the IEEE ICDM Outstanding Service Award (2017). He received the Ph.D. degree from the University of Minnesota (UMN), USA. He is a co-Editor-in-Chief of Encyclopedia of GIS, an Associate Editor of IEEE Transactions on Big Data (TBD), ACM Transactions on Knowledge Discovery from Data (TKDD), and ACM Transactions on Management Information Systems (TMIS). He has served regularly on the organization and program committees of numerous conferences, including as a Program Co-Chair of the Industrial and Government Track for the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), a Program Co-Chair for the IEEE 2013 International Conference on Data Mining (ICDM), a General Co-Chair for the IEEE 2015 International Conference on Data Mining (ICDM), and a Program Co-Chair of the Research Track for the 2018 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. He is an IEEE Fellow and an ACM Distinguished Scientist.