

STAR: A Space-Time Architecture for Semantic Trajectory Representation

Junming Shao

Big Data Research Center

University of Electronic Science and
Technology of China
Chengdu, China
junmshao@uestc.edu.cn

Ruihi Wu

Big Data Research Center
University of Electronic Science and
Technology of China
Chengdu, China
ruizhiwu@uestc.edu.cn

Weiwen Xu

Big Data Research Center

University of Electronic Science and
Technology of China
Chengdu, China
weiwen.xuu@gmail.com

Chongming Gao

Big Data Research Center
University of Electronic Science and
Technology of China
Chengdu, China
chongming.gao@gmail.com

Honglian Wang

Big Data Research Center

University of Electronic Science and
Technology of China
Chengdu, China
hlw.uestc@gmail.com

Qinli Yang

Big Data Research Center
University of Electronic Science and
Technology of China
Chengdu, China
qinli.yang@uestc.edu.cn

ABSTRACT

Trajectory data has been generated at an explosive rate in recent years due to the increasing use of location-aware devices. To mine trajectory data, a critical task is to learn a good trajectory representation since it is essential to downstream mining tasks such as trajectory retrieval and clustering. Established representation approaches tend to ignore the temporal information carried in trajectories, or treat spatial and temporal information separately. However, such strategy may be hard to reflect the reality. Beyond, the semantic information in trajectories is insufficiently exploited and enriched in existing approaches. In this paper, we propose a Space-Time Architecture for semantic trajectory Representation, called **STAR**, which simultaneously learns the space and time information carried in trajectories with a united framework. The basic idea is to learn trajectory representation as a bridge to connect location and time information as a whole, by predicting the next visiting location and time via distributed vector representation and adaptive Hawkes process. More importantly, the semantics, are further learned during the trajectory distributed vector representation process in an unsupervised way. Extensive experiments on several real-world trajectory data sets show that our proposed framework not only allows learning a good representation to capture the spatial and temporal information effectively, but also has good performance of semantic enrichment.

CCS CONCEPTS

- **Information systems** → *Location based services;*

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

KDD 2018, February 2018, London, United Kingdom

© 2018 Copyright held by the owner/author(s).

ACM ISBN 123-4567-24-567/08/06...\$15.00

<https://doi.org/10.1145/nnnnnnnn.nnnnnnn>

KEYWORDS

Trajectory representation, space-time, semantic enrichment

ACM Reference Format:

Junming Shao, Weiwen Xu, Honglian Wang, Ruihi Wu, Chongming Gao, and Qinli Yang. 2018. STAR: A Space-Time Architecture for Semantic Trajectory Representation. In *Proceedings of ACM SIGKDD (KDD 2018)*. ACM, New York, NY, USA, Article 4, 9 pages. <https://doi.org/10.1145/nnnnnnnn.nnnnnnn>

1 INTRODUCTION

With the increasing use of location-aware devices, a large volume of trajectory data has been generated to capture the movement of objects (e.g., people, vehicles, animals and hurricanes) in space-time. Mining these trajectory data is of significant importance since it allows discovering the underlying moving patterns and plays an important role in a wide range of practical applications, such as urban computing, travel recommendations and location-based services.

However, due to the different sampling rates and varying lengths of trajectories in real-world applications, the first step of extracting knowledge from trajectory data often needs to find a good trajectory representation. To date, there are mainly three strategies: point-based representation [2, 11, 30, 32], segment-based representation [6, 15–17] and feature-based representation [3, 5, 22]. The point-based representation aims to identify the most important key points in each trajectory, and use these points to represent the trajectory information, such as stay points [33] and stops [2]. Segment-based representation is to partition a trajectory into several parts, and use these parts to represent a trajectory. Examples include MDL-based trajectory segmentation [17], walk and non-walk segments [8], and roadmap-guided segmentation [23]. Feature-based methods aim to extract some geographic and/or geometric properties to characterize a trajectory, ranging from the simple descriptors (e.g., velocity, angle, shape, etc.) to complex descriptors (e.g., Fourier descriptors [4]). However, although semantic enrichment has been implicitly included in some existing trajectory representation approaches (e.g., stops or walk segments), they often focus on geographic and/or geometric information of trajectories, and the embedded semantic

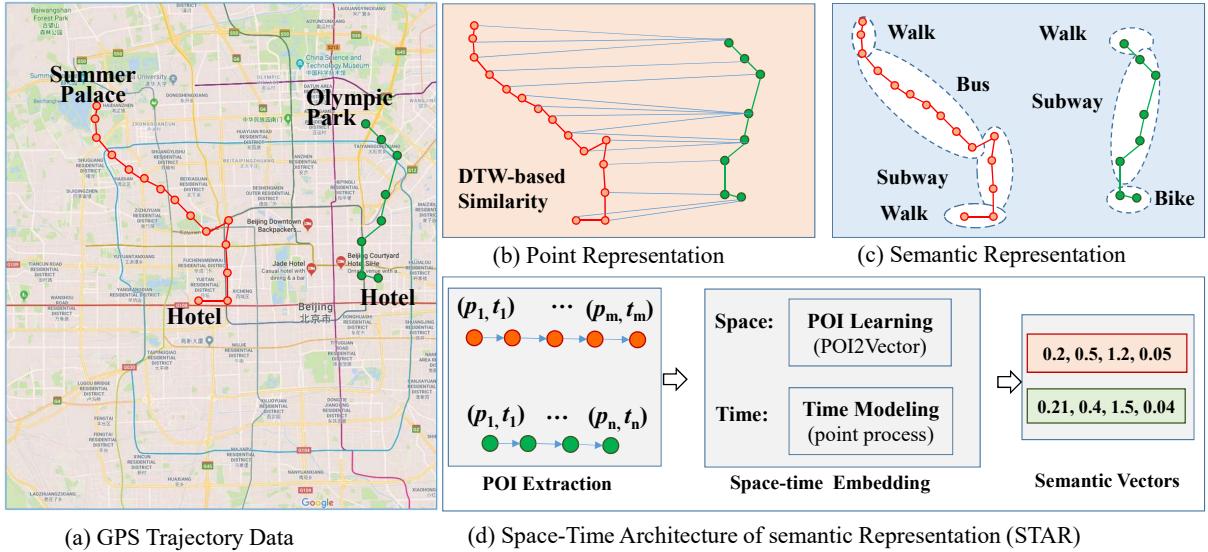


Figure 1: How to derive a good trajectory representation for downstream spatio-temporal data mining? (a) A toy example with two trajectories, where each circle indicates a GPS point. (b) The similarity is measured with the original point representation, where the semantic information is not well considered. (c) Simple semantics are learned by splitting the whole trajectory into several segments, where each segment is interpreted as "walk" or "non-walk". (d) An ideal trajectory representation by considering both spatial and temporal information, with enriched semantics.

information is insufficiently exploited. For example, Fig. 1(a) gives two travel lines in Beijing. Based on traditional trajectory representation, the two trajectories are dissimilar according to some similarity metrics (e.g., dynamic time warping (DTW) in Fig. 1(b)) since they are far apart by considering their geographic locations and geometric shapes. Fig. 1(c) further illustrates an example for simple trajectory semantic representation. However, such semantic enrichment is too coarse to capture the real semantic similarity. Beyond, established representation approaches tend to ignore the temporal information carried in trajectories, or treat spatial and temporal information separately. Thereby, deriving a new trajectory representation to characterize both spatial and temporal information with enriched semantics is of significant importance.

In this paper, we introduce a new trajectory representation method, called STAR, which aims at learning a semantic vector to simultaneously capture the spatial and temporal information carried in trajectories. Specifically, for a given trajectory data, the places of interest (POIs) are first identified. A space-time architecture is introduced to learn spatial and temporal information via distributed vector representation and adaptive Hawkes process, which allows producing a continuous semantic vector to represent each trajectory (see Fig. 1(d)). The main contributions of this paper can be summarized as follows.

- **Space-time Architecture.** Instead of focusing on spatial information and temporal information separately, we introduce a new space-time architecture for semantic trajectory representation. To the best of our knowledge, it is the first time to integrate the spatial and temporal information in a united framework. The architecture, leveraging the trajectory vector representation as a bridge, naturally integrates

spatial and temporal information as a whole (cf. Fig. 2, 4, 5 and Table 2).

- **Semantic Enrichment.** By exploiting the semantic context of POIs in the space-time architecture, STAR allows yielding continuous semantic vectors to well characterize the semantics embedded in POIs or trajectories. STAR even supports simple semantic binding (cf. Fig. 6, 7 and Table 3, 4, 5, 6).

The remainder of this paper is organized as follows: In the following section, we briefly survey related work. Section 3 presents our algorithm in detail. Section 4 contains an experimental evaluation. We finally conclude the paper in Section 5.

2 RELATED WORK

Due to the different sampling rates and varying lengths of trajectories in real-world applications, the early works of trajectory representation aim to extract key characteristic points to represent a trajectory. To define them, some concepts like stay point [18, 30, 32] and reference spots [19] are introduced. Although point representation is quite simple and intuitive, it is insufficient to characterize the whole information of a trajectory. Thereby, segment-based trajectory representation has gained growing attentions in recent years. The basic idea is to partition a trajectory into several segments and each segment is assumed to be smooth in terms of geometric shape or is enriched with specific meaning. For example, Lee et al. [16] propose a new partition strategy to split trajectory into segments with the principle of minimal description length (MDL). Zheng et al. partition trajectories into *walk segments* and *non-walk segments* based on locations and velocities of moving objects [31]. Beyond, considering the real-world constraints, such as road, Song et al. propose a roadmap-guided trajectory segmentation [24]. Another way

to characterize trajectory resorts to feature-based representation. Instead of using points or segments, feature-based representation aims to extract information from trajectory. For example, Gariel et al. [12] first sample trajectories to equal length, and then employ principal component analysis (PCA) to yield a low-dimensional vector representation. With long-short term memory (LSTM) networks, Ardakani and Hashimo learn the spatial path features. Building upon these representations, similarity metric has to be defined for subsequent mining tasks, like DTW, LCSS, ERP, SSPD. For example, dynamic time warping (DTW) distance [29] is an effective measure to quantify the similarity between two trajectories with different lengths. The longest common subsequence (LCSS) distance [26] is further proposed to eliminate the noise effect. For all established approaches, trajectories are usually studied in the space dimension, and the temporal information is little investigated. However, the time patterns carried in trajectories also provide valuable information for us. Moreover, although more and more trajectory representation approaches have shifted to trajectory semantic enrichment, such as start points, walk points, drive points and stop points in point-based representation, or walk and non-walk segments in segment-based approaches, the semantics are too specific and coarse to fully represent a trajectory. Therefore, in this paper, we propose a new space-time architecture to learn a semantic vector representation to capture spatial and temporal information in a unified framework.

3 STAR ALGORITHM

3.1 Overview of the Proposed Framework

As stated in Section 1, our key idea is to learn a semantic vector representation to capture the spatial and temporal patterns carried in each trajectory simultaneously. To this end, an intuitive space-time architecture is introduced by leveraging the trajectory vector representation as a bridge to connect space and time information, while the POI and trajectory vector representations are automatically learned by maximizing the prediction performance of all pairs of the next visiting location and time (See Fig. 2). The framework consists of three closely related components. (1) Space part: motivated by existing distributed vector word representation, the training objective of this part is to learn POI vector representations that are good at predicting the next POI (i.e., p_i in the yellow box of Fig. 2). Here, the trajectory vector representations in the second component (indicated as green box of Fig. 2) are also integrated. Namely, for better predicting the next visiting POI, we use both the context of the POI and its associated trajectory information. (2) Space-time trajectory representation part: the objective is to learn trajectory vector representation to help predict the location p_i and its associated time $t_i^{p_i}$ simultaneously. This part intuitively connects the space and time part as a whole. (3) By integrating the vector representation of POI and trajectory, the adaptive Hawkes process is introduced to model the past timings of visiting the POI. Thereby, by combining the three components together, the next visiting location and time can be predicted. The semantic POI and trajectory representations are learned over all trajectories. More importantly, the spatial and temporal information is well integrated automatically in the trajectory vector representation.

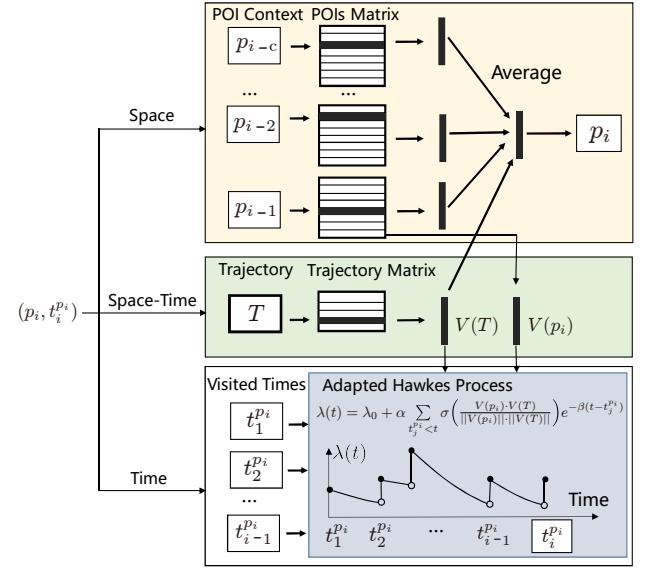


Figure 2: Illustration of proposed space-time architecture for semantic trajectory representation. The objective is to learn semantic trajectory representation to help predict the next location and time simultaneously. The framework consists of three closely related components with different colors. The POIs matrix represents the POI vector representations with the size of $N \times L$, where N is number of whole POIs, and L is the length of a vector representation. The trajectory matrix is the semantic trajectory representation for all trajectories with size of $M \times L$, where M is the number of trajectories. p_{i-1}, \dots, p_{i-c} is the context of POI p_i , \mathcal{T} is the associated trajectory of p_i , $V(\mathcal{T})$ and $V(p_i)$ are the vector representation of the trajectory \mathcal{T} and POI p_i , respectively. $t_1^{p_i}, \dots, t_{i-1}^{p_i}$ mean all previous time points of visiting the location p_i .

Specifically, for each pair of location-time $(p_i, t_i^{p_i})$, the context of p_i representing the geometric, trajectory and semantic neighbors (i.e., p_{i-1}, \dots, p_{i-c}) is first identified, where c is the number of context neighbors. These POIs, plus the associated trajectory \mathcal{T} of p_i are used to predict p_i . On the other side, all previous visiting timings of p_i (i.e., $t_1^{p_i}, \dots, t_{i-1}^{p_i}$) are used to model an adaptive Hawkes process, and then we use it to maximize the predicting performance of the next visiting time $t_i^{p_i}$ (see Fig. 2).

3.2 Distributed Vector Representation

Distributed vector representation [7] is originated in the field of natural language processing, and has been demonstrated as an efficient way to capture a large number of precise syntactic and semantic word relationships. To date, one of the most widely used model is Continuous Bag-Of-Words (CBOW) [25]. The training objective of the CBOW model is to learn word representations that are used to predict the current word with its surrounding words in a sentence.

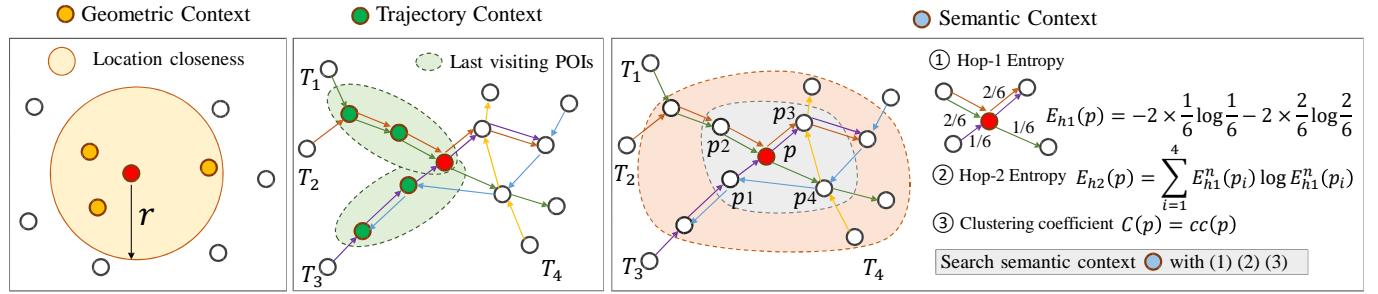


Figure 3: Illustration of the context of a given POI. The context of p is determined from three aspects: geometric context, trajectory context and semantic context. Geometric context means the nearest POIs in the geometric space. Trajectory context means the last visiting POIs before p_i . Semantic context indicates the POIs with similar semantic information. Here similar semantic information is evaluated based on its local connectivity pattern, which are quantified by its hop-1 entropy, hop-2 entropy and its clustering coefficient. $E_{h1}^n(p_i)$ indicates the normalized hop-1 entropy.

In the context of trajectory data, the context of a given POI is constrained by its exact location. Intuitively, POIs can be analogous to words while trajectories are analogous to sentences. However, they differ largely since two POIs (or trajectories) with similar semantic may not adjoin each other. Therefore, the search of context of a given POI should be carefully examined. Here, we introduce three types of context information to capture both geometric and semantic patterns for a given POI, which are geometric context, trajectory context and semantic context. The geometric context is represented as its nearest location neighbors (see Fig. 3(a)). The trajectory context is used to capture the routing patterns in trajectories. Namely, the trajectory context of a given POI is its last visited POIs. Considering the spatial dependence in human or animal movement, two last visiting POIs are considered as its nearest neighbors in term of trajectory context. The semantic context, which means the POIs sharing similar semantic information, is captured by searching similar connectivity patterns. The basic idea is quite intuitive: when two POIs share similar semantic information, their local connectivity patterns tend to be similar (e.g., two train stations may show similar linking patterns with other POIs). Therefore, motivated by natural observations, we report three measures to capture the semantic context, which are hop-1 entropy, hop-2 entropy and clustering coefficient.

Formally, for a given POI p_i , let Γ be its directly connecting neighbor set, its hop-1 entropy $E_{h1}(p_i)$ is defined as follows.

$$E_{h1}(p_i) = - \sum_{p_j \in \Gamma} \mathcal{P}(p_j) \log_2 \mathcal{P}(p_j) \quad (1)$$

where $\mathcal{P}(p_j) = \frac{c(p_i, p_j)}{\sum_{p_j \in \Gamma} c(p_i, p_j)}$, $c(p_i, p_j)$ means the number of connections between p_i and p_j . Hop-1 entropy captures the local distribution of connection strength with its hop-1 neighbors for a given POI. The hop-2 entropy $E_{h2}(p_i)$ is built upon the hop-1 entropy, which is defined as follows.

$$E_{h2}(p_i) = - \sum_{p_j \in \Gamma} E_{h1}^n(p_j) \log_2 E_{h1}^n(p_j) \quad (2)$$

Where $E_{h1}^n(p_j)$ is the normalized hop-1 entropy of p_j . Therefore, hop-2 entropy further captures the whole distribution of connecting strength in the range of hop-2 neighbors. Beyond, considering the

connectivity density, the well-known clustering coefficient [27] (i.e., $cc(p_i)$) is calculated. Finally, for each p_i , three measures $E_{h1}(p_i)$, $E_{h2}(p_i)$ and $cc(p_i)$ are obtained. The semantic neighbors of p_i are thus searched by finding similar POIs in terms of the three measures. For illustration, Fig. 3 demonstrates how to identify geometric, trajectory and semantic context, respectively.

Building upon the context, we can learn the distributed vector representation of POIs and trajectories with CBOW model. Formally, given a p_i , the objective of the CBOW model is to maximize the average logarithm probability as follows.

$$\sum_{i=1}^N \log \mathcal{P}(p_i | context(p_i)) \quad (3)$$

where $context(p_i) = \{p_{i-1}, \dots, p_{i-c}, \mathcal{T}\}$ is the context of p_i , \mathcal{T} is the associated trajectory of p_i , N is the number of all POIs for a given trajectory data set. The $\mathcal{P}(p_i | context(p_i))$ is a softmax function. By considering the trajectory information, the Eq. (3) is further written as follows.

$$\mathcal{P}(p_i | context(p_i)) = \frac{\exp(V(context(p_i))' V(p_i))}{\sum_{k=1}^N \exp(V(context(p_k))' V(p_k))} \quad (4)$$

where $V(p_i)$ is the distributed vector representation of p_i , and $V(X_{p_i}) = V(context(p_i)) = \frac{1}{c+1} \left(\sum_{1 \leq j \leq c} V(p_{i-j}) + V(\mathcal{T}) \right)$.

However, since N is usually large in real-world scenarios, the hierarchical softmax [21] is thus employed to speed up the training process. More precisely, each POI (e.g., p_i) can be reached by an appropriate path from the root of a binary Huffman tree [20]. Let $n(p_i, j)$ be the j -th node on the path from the root to p_i , and let $L(p_i)$ be the length of this path, so $n(p_i, 1) = \text{root}$ and $n(p_i, L(p_i)) = p_i$. In addition, for any inner node n , let $ch(n)$ be an arbitrary fixed child of n and let $[x]$ be 1 if x is true and -1 otherwise. Then the hierarchical softmax defines $\mathcal{P}(p_i | context(p_i))$ as follows:

$$\begin{aligned} \mathcal{P}(p_i | context(p_i)) &= \prod_{k=1}^{L(p_i)-1} \sigma([n(p_i, k+1) = ch(n(p_i, k))] \\ &\quad \cdot V(n(p_i, k))' V(context(p_i))) \end{aligned} \quad (5)$$

where $\sigma(x) = \frac{1}{1+exp(-x)}$.

Finally, the training objective function is to maximize the log-likelihood as follows.

$$\begin{aligned} \mathcal{L}_s(p_i|\theta_{p_i}^s) &= \sum_{k=1}^{L(p_i)-1} \log \left(\sigma \left([n(p_i, k+1) \right. \right. \\ &\quad \left. \left. - ch(n(p_i, k))] \cdot V(n(p_i, k))' V(context(p_i)) \right) \right) \end{aligned} \quad (6)$$

where $\theta_{p_i}^s = \{V(p_i-j), V(\mathcal{T}), V(n(p_i, k)\}$, and $j = 1, \dots, c$.

3.3 Time Modeling via Temporal Point Process

To derive a better trajectory representation, time information in trajectories needs to be considered. However, unlike spatial information, the temporal information characterizes the trajectory in another dimension. It is a non-trivial task to couple space and time information directly. Thereby, as stated before, we use the trajectory vector representation as a bridge to connect spatial and temporal information. Namely, we integrate the trajectory and POI space information to help time modeling.

To exploit the temporal patterns in each trajectory, point process is employed in this study. Point process models are useful for describing events occurring at random locations and times. It has gained growing attentions in different fields, especially in recommendation community [9, 10, 14, 28]. To characterize temporal point processes, the most intuitive way is to resort to the conditional intensity function $\lambda(t|\mathcal{H}_t)$, which is defined as the expected number of events per unit of time given all previous events \mathcal{H}_t .

$$\begin{aligned} \lambda(t|\mathcal{H}_t) &= \lim_{dt \rightarrow 0} \frac{\mathbb{E}[\mathcal{N}(t+dt) - \mathcal{N}(t)|\mathcal{H}_t]}{dt} \\ &= \lim_{dt \rightarrow 0} \frac{\mathcal{P}[\mathcal{N}(t+dt) - \mathcal{N}(t) > 0|\mathcal{H}_t]}{dt} \end{aligned} \quad (7)$$

For simplicity, we use $\lambda^*(t)$ to denote the conditional intensity function. Therefore, $\lambda^*(t)dt$ characterizes the probability for the occurrence of a new event given the history \mathcal{H}_t , i.e.,

$$\lambda^*(t)dt = \mathcal{P}(\text{a new event occurs in } [t, t+dt] | \mathcal{H}_t) \quad (8)$$

In addition, given a sequence of events $T = \{t_1, \dots, t_n\}$, supposing that the conditional density function is $f^*(t)$, we have the following equation [1, 10].

$$\lambda^*(t)dt = \frac{f^*(t)dt}{S^*(t)} = \frac{f^*(t)dt}{1 - F^*(t)}, \quad (9)$$

where $S^*(t) = \exp \left(- \int_{t_n}^t \lambda^*(\tau)d\tau \right)$ is the probability that no new event occurs from last event time t_n to the current time t , and $F^*(t)$ is the cumulative probability that a new event will happen before time t since the last event time t_n . Finally, given a sequence of events $\{t_1, \dots, t_n\}$, the likelihood can be expressed as follows.

$$\mathcal{L}(\{t_1, \dots, t_n\}) = \prod_{i=1}^n \lambda^*(t_i) \exp \left(- \int_{t_{i-1}}^{t_i} \lambda^*(\tau)d\tau \right) \quad (10)$$

Therefore, the critical point to characterize temporal point processes is to design a specific conditional intensity function $\lambda^*(t)$ to capture the temporal patterns in real-world applications (e.g., $\lambda^*(t) = \lambda_0$ corresponds to a homogeneous Poisson process). In this study, we introduce an adaptive Hawkes process to model the

moving patterns existing in trajectories. The Hawkes process, or known as the self-exciting process, is an extension of the Poisson processes that aims to explain excitatory interactions [13]. Since the events in a past interval can affect the occurrence of the events in later intervals, the Hawkes process in general is more expressive than traditional Poisson process. The basic form of Hawkes process with its conditional intensity function is as follows.

$$\lambda^*(t) = \lambda^0 + \alpha \sum_{i=1}^n g(t, t_i), \quad (11)$$

where $g(t, t_i) \geq 0$ is a kernel function expressing the positive influence of past events on the current event. The most widely used kernel function is the response function $g(t, t_i) = \exp(-\beta(t-t_i))$. In this study, we reformat the Hawkes process by introducing the spatial information (i.e., the distributed vector representation $V(\mathcal{T})$ and $V(p_i)$) as follows.

$$\lambda_{p_i}^*(t|\theta_{p_i}^t) = \lambda_{p_i}^0 + \alpha_{p_i} \cdot \sigma \left(\frac{V(\mathcal{T})' V(p_i)}{\|V(\mathcal{T})\| \cdot \|V(p_i)\|} \right) \sum_{i=1}^n \exp(-\beta_{p_i}(t-t_i)), \quad (12)$$

where $\theta_{p_i}^t = \{\lambda_{p_i}^0, \alpha_{p_i}, \beta_{p_i}, V(\mathcal{T}), V(p_i)\}$ is the parameter set. $\lambda_{p_i}^0$ is the base intensity of visiting p_i , and $\sigma \left(\frac{V(\mathcal{T})' V(p_i)}{\|V(\mathcal{T})\| \cdot \|V(p_i)\|} \right)$ is used to integrate the space-time information into the temporal point process. If the distributed vector representation of POI (i.e., $V(p_i)$) is similar with the vector representation of trajectory (i.e., $V(\mathcal{T})$), it means there is a higher probability that the p_i will be visited and thus α_{p_i} is enhanced in iterations. It is worth mentioning that the p_i of each trajectory has different intensity function $\lambda_{p_i}^*(t|\theta_{p_i}^t)$, and we learn it for each trajectory, respectively.

Finally, given the occurrence observation t_1, \dots, t_n for an interval $[0, t](t \geq t_n)$, and with the conditional intensity function $\lambda_{p_i}^*(t|\theta)$, the log-likelihood of the adaptive Hawkes process is given as follows.

$$\begin{aligned} \mathcal{L}_t(\{t_1, \dots, t_n\}|\theta_{p_i}^t) &= \sum_{i=1}^n \frac{\alpha_{p_i} \cdot C(p_i, \mathcal{T})}{\beta_{p_i}} [\exp(-\beta_{p_i}(t_n - t_i)) - 1] \\ &\quad + \sum_{i=1}^n \log [\lambda_{p_i}^0 + \alpha_{p_i} \cdot C(p_i, \mathcal{T}) \cdot A_{p_i}(i)] - \lambda_{p_i}^0 t_n \end{aligned} \quad (13)$$

where $A_{p_i}(i) = \sum_{t_j < t_i} \exp(-\beta_{p_i}(t_i - t_j))$ for $i \geq 2$, and $A_{p_i}(1) = 0$.

$$C(p_i, \mathcal{T}) = \sigma \left(\frac{V(\mathcal{T})' V(p_i)}{\|V(\mathcal{T})\| \cdot \|V(p_i)\|} \right).$$

3.4 Parameter Learning

Given a trajectory collection \mathbb{T} , $\mathcal{T} \in \mathbb{T}$, where each trajectory $\mathcal{T} = \bigcup_{i=1}^{n_i} (p_i, t_i^{p_i})$, n_i is the number of POIs in \mathcal{T} . By exploiting both spatial and temporal information, the objective function is to maximize Eq. (6) and Eq. (13), simultaneously.

$$\mathcal{L} = \arg \min_{\theta_{p_i}^s, \theta_{p_i}^t} \sum_{i=1}^N (\mathcal{L}_s(p_i|\theta_{p_i}^s) + \mathcal{L}_t(\{t_1, \dots, t_n\}|\theta_{p_i}^t)) \quad (14)$$

Since the objective function is smooth and convex, and thus the gradient descent can be employed directly. Finally, we can obtained

Algorithm 1 Pseudocode of STAR algorithm

Input: Trajectory data: \mathbb{T} , POI set: \mathbb{P}
Parameters: Vector length l , Number of neighbors c

Output: Each $V(\mathcal{T})$ and $V(p_i)$, $\mathcal{T} \in \mathbb{T}$ and $p_i \in \mathbb{P}$.

```

1: while not converges do
2:   for each trajectory  $\mathcal{T} \in \mathbb{T}$  do
3:     for each  $p_i$  in  $\mathcal{T}$  do
4:       Extract all previous times  $(t_1^{p_i}, \dots, t_n^{p_i})$  of visiting  $p_i$ 
5:       //Context Search
6:       Obtain the geometric context of  $p_i$ 
7:       Obtain the trajectory context of  $p_i$ 
8:       Search the semantic context of  $p_i$  via Eqs. (1) and (2)
9:       //Parameter learning
10:      Update the vector  $V(\omega_{p_i}^{j-1})$  via Eq. (15)
11:      Update the vector  $V(X_{p_i})$  via Eq. (16)
12:      Update the vector  $V(p_i)$  via Eq. (17)
13:      Update the vector  $V(\mathcal{T})$  via Eq. (18)
14:      Update the parameter  $\alpha_{p_i}$  via Eq. (19)
15:      Update the parameter  $\beta_{p_i}$  via Eq. (20)
16:      Update the parameter  $\lambda_{p_i}^0$  via Eq. (21)
17:    end for
18:  end for
19: end while

```

the update rules as follows.

$$V(\omega_{p_i}^{j-1}) := V(\omega_{p_i}^{j-1}) + \eta \left[1 - d_{p_i}^j - \sigma(V(X_{p_i})' V(\omega_{p_i}^{j-1})) \right] V(X_{p_i}) \quad (15)$$

where $\omega_{p_i}^j$ is the vector representation of j -th node of the path in the Huffman tree to p_i , and $d_{p_i}^j \in \{0, 1\}$ is the Huffman code of p_i .

$$V(X_{p_i}) := V(X_{p_i}) + \eta \sum_{j=2}^{L(p_i)} \left[1 - d_{p_i}^j - \sigma(V(X_{p_i})' V(\omega_{p_i}^{j-1})) \right] V(\omega_{p_i}^{j-1}) \quad (16)$$

$$\begin{aligned} V(p_i) := & V(p_i) + \eta \left(\alpha_{p_i} \sum_{j=1}^n C(p_i, \mathcal{T}) (1 - C(p_i, \mathcal{T})) \frac{V(\mathcal{T})'}{\|V(\mathcal{T})\|} \right. \\ & \cdot \left(\frac{1}{\|V(p_i)\|} - \frac{V(p_i)^2}{\|V(p_i)\|^3} \right) \left(\frac{\exp(-\beta_{p_i}(t_n - t_j)) - 1}{\beta_{p_i}} \right. \\ & \left. \left. + \frac{A_{p_i}(j)}{\lambda_{p_i}^0 + \alpha_{p_i} C(p_i, \mathcal{T}) A_{p_i}(j)} \right) \right) \end{aligned} \quad (17)$$

The update rule of $V(\mathcal{T})$ is:

$$\begin{aligned} V(\mathcal{T}) := & V(\mathcal{T}) + \eta \left(\sum_{j=2}^{L(p_i)} \left[1 - d_{p_i}^j - \sigma(V(X_{p_i})' V(\omega_{p_i}^{j-1})) \right] V(X_{p_i}) \right. \\ & \left. + \alpha_{p_i} \sum_{k=1}^n C(p_i, \mathcal{T}) (1 - C(p_i, \mathcal{T})) \left(\frac{1}{\|V(\mathcal{T})\|} - \frac{V(\mathcal{T})^2}{\|V(\mathcal{T})\|^3} \right)' \right) \end{aligned}$$

$$\cdot \frac{V(p_i)}{\|V(p_i)\|} \left(\frac{\exp(-\beta_{p_i}(t_n - t_k)) - 1}{\beta_{p_i}} + \frac{A_{p_i}(k)}{\lambda_{p_i}^0 + \alpha_{p_i} C(p_i, \mathcal{T}) A_{p_i}(k)} \right) \quad (18)$$

The update rule for parameters of Hawkes process of α_{p_i} , β_{p_i} and $\lambda_{p_i}^0$ are as follows.

$$\begin{aligned} \alpha_{p_i} := & \alpha_{p_i} + \eta \left(C(p_i, \mathcal{T}) \sum_{k=1}^n \left(\frac{\exp(-\beta_{p_i}(t_n - t_k)) - 1}{\beta_{p_i}} \right. \right. \\ & \left. \left. + \frac{A_{p_i}(k)}{\lambda_{p_i}^0 + \alpha_{p_i} C(p_i, \mathcal{T}) A_{p_i}(k)} \right) \right) \end{aligned} \quad (19)$$

$$\begin{aligned} \beta_{p_i} := & \beta_{p_i} - \eta \left(\alpha_{p_i} C(p_i, \mathcal{T}) \sum_{k=1}^n \left(\frac{t_n - t_k}{\beta_{p_i}} \exp(-\beta_{p_i}(t_n - t_k)) \right. \right. \\ & \left. \left. + \frac{\exp(-\beta_{p_i}(t_n - t_k)) - 1}{\beta_{p_i}^2} + \frac{B_{p_i}(k)}{\lambda_{p_i}^0 + \alpha_{p_i} C(p_i, \mathcal{T}) A_{p_i}(k)} \right) \right) \end{aligned} \quad (20)$$

$$\lambda_{p_i}^0 := \lambda_{p_i}^0 + \eta \left(-t_n + \sum_{k=1}^n \frac{1}{\lambda_{p_i}^0 + \alpha_{p_i} C(p_i, \mathcal{T}) A_{p_i}(k)} \right) \quad (21)$$

Finally, the pseudocode of STAR is given in Algorithm 1.

4 EXPERIMENT

4.1 Experimental Setup

Data set. Here, to evaluate the performance of STAR, five real-world trajectory data sets are used. Specifically, **Beijing taxi data** contains more than 100 thousand taxi trajectories in Beijing over the period from 2008/2/2 to 2008/2/8. **Geolife data** was collected by Microsoft Research Asia. It includes 179 users life trajectory data in five years (from April 2007 to August 2012). This data set is benefit to analyzing user's moving pattern in a long time. **Animal movement dataset** contains the radio-telemetry locations (with other information) of Homing pigeons, Brown pelican and Swainson Hawk over the years 2011 through 2016. For more details, please refer to the website (<https://www.movebank.org/>). This data set is applied to study animal behavior in wild. Table 1 further lists the statistics of these trajectory data sets.

POIs Extraction. To learn the vector representation, we first partition the study area of five data sets into $n \times n$ grid cells ($n = 100$ in this study). Afterwards, the average longitude and latitude of a grid of GPS points and time information are extracted, and the pair-wise similarity matrix among cells is computed. Finally, we

Table 1: Statistics of five trajectory data sets.

Dataset	#object	#GPS points	Time Start	Time End
Geolife	179	24,876,978	2007/4/1	2012/8/1
Beijing Taxi	10272	17,662,984	2008/2/2	2008/2/8
Homing pigeons	35	6,229,315	2013/7/29	2014/9/17
Brown pelican	81	168,041	2013/4/24	2016/2/5
Swainson Hawk	17	18,828	2011/6/17	2013/8/28

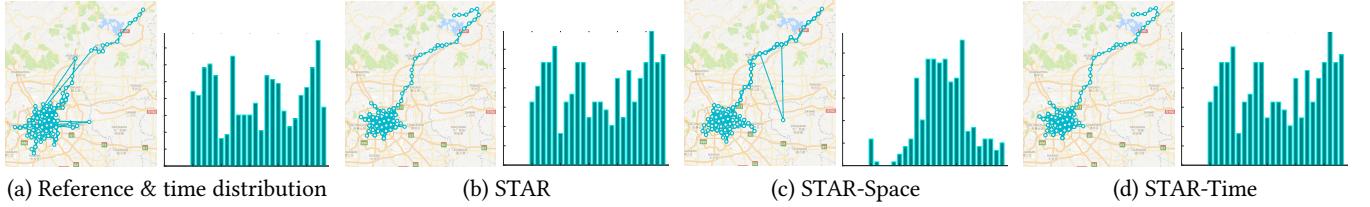


Figure 4: Illustration of space-time trajectory Representation. Here the most similar trajectory is searched by STAR with different ways including space-time, space and time, respectively.

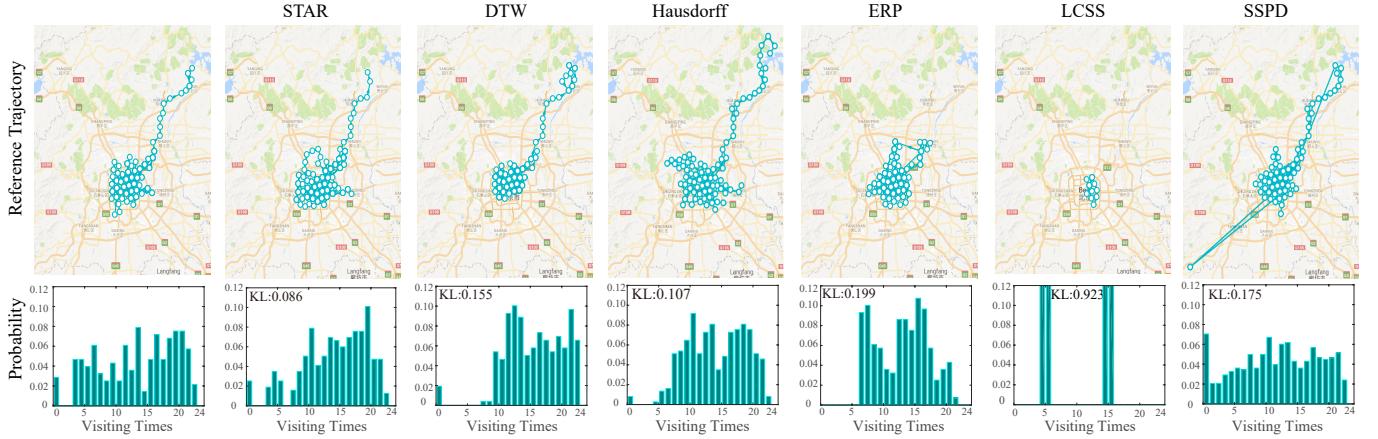


Figure 5: Illustration of the performance of trajectory retrieval with different similarity metrics. Here the trajectory locations and its visiting time distributions are plotted.

Table 2: The comparsion of time distribution similarity for top-1 trajectory with different similarity measures.

	STAR	DTW	Hausdorff	ERP	LCSS	SSPD
Taxi	0.104±.03	0.118±.03	0.110±.02	0.163± .05	0.142±.06	0.116±.03
Geolife	0.437	0.446	0.465	0.456	0.463	0.451
Hawk	1.517	1.683	2.071	1.919	2.532	2.111
Pigeon	0.077	1.356	3.268	4.243	0.448	1.719
Pelican	1.692	2.160	1.870	1.700	1.967	1.894

employ K-means clustering method to extract k number of POIs ($k=1000$ for all data sets in this study).

Semantic Ground-truth Generation. To evaluate the performance of trajectory enrichment, we manually labeled the semantic information for each POI on Beijing Taxi data and Geolife data since we can perform semantic enrichment for POIs via google and Baidu maps. We divide these POIs into eight semantic categories, including scenic area, park, village, community, transportation, business area, hospital and education. Moreover, the semantics of all POIs for the two data sets, the sourcecode of STAR and the five real-world data sets are all publicly available at: <http://dm.uestc.edu.cn/star/>.

Evaluation Matrices. Here the $\text{Acc}@k$ is used to evaluate the POI semantic enrichment, which is defined as the percentage of same semantic POIs by search top- k POIs. The Kullback-Leibler divergence (KL) is employed to measure the similarity of time distribution and trajectory semantic distribution.

4.2 Space-time Representation Evaluation

Here, we first check whether STAR allows capturing spatial and temporal information effectively. For better illustration, a reference trajectory is randomly selected from taxi data set, and its time distribution is also recorded (see Fig. 4(a)). Fig. 4(b)-(d) plot the most similar trajectory searched by STAR with different strategies, including space-time representation (STAR), space representation (STAR-space, with space information only) and time representation (STAR-Time, with temporal information only), respectively. From these figures, we can observe that STAR allows searching trajectory with both high space and time similarity. STAR-Space focuses on more space similarity while the time information cannot be captured. For STAR-Time, it yields the same result of STAR, which further demonstrated the time patterns are well learned.

Beyond, to qualitatively evaluate our algorithm, we further compare it to several similarity metrics paradigms, including DTW, Hausdorff, ERP, LCSS and SSPD. We perform the trajectory retrieval task on all data sets, and the similarity of time distribution is evaluated based on Kullback-Leibler divergence. Since it is time consuming to compute all these comparing metrics on taxi data (more than one day for each metric), 100 randomly trajectories are selected and we repeat it ten times. Fig. 5 plots the performance of trajectory retrieval with different similarity metrics for a reference trajectory. Table 2 further summarizes the performances of seven algorithms on the five real-world data sets. From the table, we can

Table 3: Examples of semantic similarity and semantic binding of derived POIs on Beijing taxi data set.

Type	Ref. ID	Semantic	Top-1 Semantic	Top-2 Semantic	Top-3 Semantic
Scenic Area	531	Great Wall of Badaling	956 Great Wall of Juyongguan	663 Dayangshan National Forest Park	890 Cuahuashan Natural Scenic Area
Park	430	Binshui Park	593 Beijing International Sculpture Park	408 Olympic Forest Park	520 Badachu Park
Village	591	Dongxinzhuang Village	676 Qingdian Village	4 Yangpodi Village	39 Zhaotuzhuang Village
Community	349	Chedaogou Nanli Community	908 Zhuzijie Community	628 Dayoubeili Community	785 Daoxiangyuan Community
Transportation	395	Beijing Capital International Airport	747 East Suburb Wetland Park (near the airport)	507 Qinghe Railway Station	350 Wanzhuang Railway Station
Business	128	Daimler Tower	484 Ericsson Mansion	397 Volkswagen Group	293 Huapei Office Mansion
Hospital	768	Peking Union Medical College Hospital	9 Peking University Third Hospital	293 The Second Artillery General Hospital	168 Beijing Cancer Hospital
Education	396	Beijing Institute of Technology	767 Tsinghua University	879 Chinese People's Public Security University	587 Huanghoutai Village
Semantic Binding					
$V(\text{Great Wall of Mutianyu}) - V(\text{Beijing Institute of Technology}) + V(\text{Beihang University}) = V(\text{Great Wall of Jinshanling})$					
$V(\text{Daimler Tower}) - V(\text{Badachu Park}) + V(\text{Yuyuantan Park}) = V(\text{Beijing Gaode Mansion})$					
$V(\text{Nankou Railway Station}) - V(\text{Beijing Yintai Hongye Golf Club}) + V(\text{Beijing Country Golf Club}) = V(\text{Langfang Railway Station})$					
$V(\text{Tsinghua University}) - V(\text{Peking University First Hospital}) + V(\text{Military Hospital}) = V(\text{Capital Normal University})$					
$V(\text{Beijing Cancer Hospital}) - V(\text{Beijing Capital International Airport}) + V(\text{Beijing Nanyuan Airport}) = V(\text{Peking University First Hospital})$					
$V(\text{Volkswagen Group}) - V(\text{Capital Normal University}) + V(\text{Tsinghua University}) = V(\text{Huapei Office Mansion})$					

Table 4: The semantic enrichment for all POIs and different types of POIs on Beijing taxi data set.

	All	Trans.	Scenic	Park	Busi.	Hosp.	Edu.	Com.	Village
Acc@1	0.559	0.458	0.489	0.461	0.495	1.000	0.462	0.506	0.664
Acc@3	0.536	0.465	0.479	0.447	0.432	0.800	0.333	0.449	0.646
Acc@5	0.521	0.429	0.463	0.426	0.410	0.680	0.369	0.390	0.640
Acc@7	0.491	0.375	0.432	0.382	0.378	0.600	0.308	0.340	0.622
Acc@10	0.469	0.335	0.409	0.339	0.349	0.520	0.277	0.301	0.612
Acc@15	0.444	0.286	0.384	0.296	0.309	0.453	0.236	0.256	0.596
Acc@20	0.433	0.266	0.378	0.272	0.287	0.380	0.208	0.232	0.592

Table 5: The semantic enrichment for all POIs and different types of POIs on Geolife data set.

	All	Trans.	Scenic	Park	Busi.	Hosp.	Edu.	Com.	Village
Acc@1	0.529	0.418	0.391	0.474	0.451	0.500	0.597	0.467	0.687
Acc@3	0.524	0.491	0.423	0.452	0.412	0.500	0.552	0.402	0.706
Acc@5	0.517	0.498	0.405	0.432	0.424	0.350	0.531	0.386	0.707
Acc@7	0.498	0.455	0.385	0.425	0.381	0.393	0.501	0.385	0.689
Acc@10	0.478	0.438	0.363	0.404	0.355	0.450	0.460	0.367	0.675
Acc@15	0.460	0.400	0.338	0.385	0.327	0.450	0.435	0.359	0.659
Acc@20	0.446	0.381	0.321	0.372	0.307	0.425	0.404	0.347	0.649

notice that STAR outperforms all comparing algorithms, which further demonstrates its effectiveness of space-time modeling.

4.3 Semantic Enrichment Evaluation

4.3.1 POI vector representation. Here, we examine the semantic POI vector representation and its semantic binding. As stated, we

Table 6: Trajectory semantic enrichment on Geolife and taxi data sets. Here the semantic similarities with top@k trajectories are computed via Kullback-Leibler divergence.

	KL@1	KL@3	KL@5	KL@10	KL@15	KL@20
Geolife	0.167	0.188	0.202	0.223	0.235	0.245
Taxi	0.126	0.130	0.132	0.137	0.139	0.142

manually labeling all 1000 POIs for Beijing taxi data set and Geolife data set, respectively. Table 3 gives eight types of semantic POIs, and for each type, one reference POI is selected. The top three similar POIs are given. We search in the vector space for the POIs closest to the reference POI measured by Euclidean distance. From the table, we can observe that our POI vector representation allows capturing effective semantic information. Amazingly, the derived POI vector representations even support semantic binding with some simple operations such as addition and subtraction (see Table 3). For instance, the resulting vector of a scenic area adds a university, and then subtracts another university is most similar with the vector of a scenic area (e.g., $V(\text{Great Wall of Mutianyu}) - V(\text{Beijing Institute of Technology}) + V(\text{Beihang University}) = V(\text{Great Wall of Jinshanling})$). Table 4 and Table 5 further summarize the average retrieval accuracies for all POIs as well as different types of POIs on Beijing taxi and Geolife data sets by searching top@ k POIs.

4.3.2 Trajectory vector representation. Like POI vector representation evaluation, we examine the trajectory semantic enrichment. Fig. 6 plots two most similar trajectories. From the two plots, we can see the two trajectories are similar since they have many common

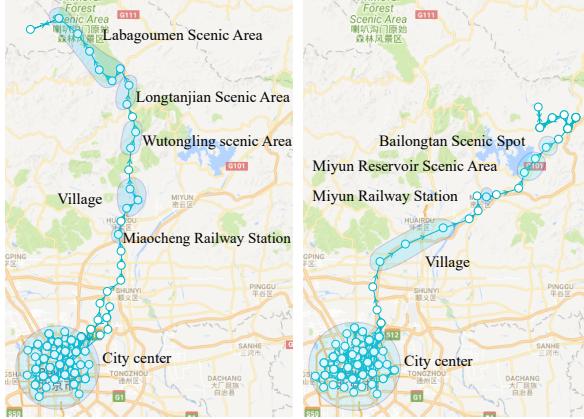


Figure 6: Illustration of trajectory semantic enrichment.

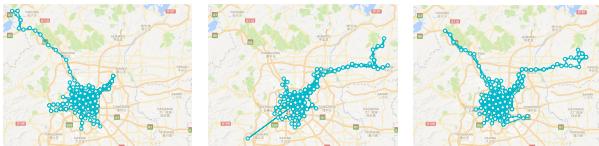


Figure 7: Illustration of trajectory semantic binding with STAR. Here the vector of left trajectory plus the vector of middle trajectory produces a new vector which is most similar to the right trajectory.

semantic POIs. Fig. 7 further gives an example for trajectory semantic binding, where the semantic vectors of two trajectories can be added to produce a new trajectory, which resembles their semantic summation. To further evaluate trajectory semantic enrichment, a trajectory retrieval task is performed, where top@ k trajectories are found via our derived trajectory semantic vectors by regarding each trajectory as reference. Table 6 gives the average semantic similarities of the retrieval results with KL divergence.

5 CONCLUSION

In this paper, we introduce a space-time architecture for trajectory semantic representation. In contrast to existing approaches, STAR allows producing a continuous semantic vector to capture both spatial and temporal information carried in trajectories. More importantly, the semantic is well learned. To this end, distributed vector representation and temporal point process are employed. Although many improvements can be enhanced, the current study on extensive experiments have demonstrate its benefits in both space-time modeling and semantic enrichment.

REFERENCES

- [1] Odd Aalen, Ornulf Borgan, and Hakon Gjessing. 2008. *Survival and event history analysis: a process point of view*. Springer Science & Business Media.
- [2] Luis Otavio Alvares, Vania Bogorny, Bart Kuijpers, Jose Antonio Fernandes de Macedo, Bart Moelans, and Alejandro Vaisman. 2007. A model for enriching trajectories with semantic geographical information. In *Proceedings of the 15th annual ACM international symposium on Advances in geographic information systems*. ACM, 22.
- [3] Shin Ando, Theerasak Thanomphongphan, Yoichi Seki, and Eimoshin Suzuki. 2015. Ensemble anomaly detection from multi-resolution trajectory features. In *Data Mining and Knowledge Discovery*.
- [4] Ronald Annini and Carlos H Q Forster. 2012. Analysis of Aircraft Trajectories using Fourier Descriptors and Kernel Density Estimation. In *ITSC'12*.
- [5] Ilya S Ardkani and Koichi Hashimoto. 2017. Encoding Bird's Trajectory Using Recurrent Neural Networks. In *ICMA'17*.
- [6] Richard Bellman. 1961. On the Approximation of Curves by Line Segments using Dynamic Programming. In *Communications of the ACM*.
- [7] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research* 3, Feb (2003), 1137–1155.
- [8] Yukun Chen, Kai Jiang, Yu Zheng, Chunping Li, and Nenghai Yu. 2009. Trajectory simplification method for location-based social networking services. In *Proceedings of the 2009 International Workshop on Location Based Social Networks*. ACM, 33–40.
- [9] Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song. 2016. Recurrent marked temporal point processes: Embedding event history to vector. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1555–1564.
- [10] Nan Du, Yichen Wang, Niao He, Jimeng Sun, and Le Song. 2015. Time-sensitive recommendation from recurrent user activities. In *Advances in Neural Information Processing Systems*. 3492–3500.
- [11] Renato Filiotto, Vania Bogorny, Cleto May, and Douglas Klein. 2015. Semantic enrichment and analysis of movement data: probably it is just starting! *SIGSPATIAL Special* 7, 1 (2015), 11–18.
- [12] Maxime Gariel, Ashok N Srivastava, and Eric Feron. 2011. Trajectory Clustering and An Application to Airspace Monitoring. In *Transactions on Intelligent Transportation Systems*.
- [13] Alan G Hawkes. 1971. Spectra of some self-exciting and mutually exciting point processes. *Biometrika* 58, 1 (1971), 83–90.
- [14] Seyed Abbas Hosseini, Keivan Alizadeh, Ali Khodadadi, Ali Arabzadeh, Mehrdad Farajtabar, Hongyan Zha, and Hamid R Rabiee. 2017. Recurrent poisson factorization for temporal recommendation. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 847–855.
- [15] Jinseng Jiang, Chong Xu, Jian Xu, Ming Xu, Ning Zheng, and Kaiwei Kong. 2016. Route planning for locations based on trajectory segments. In *Proceedings of the 2nd ACM SIGSPATIAL Workshop on Smart Cities and Urban Analytics*.
- [16] Jae-Gil Lee, Jiawei Han, and Kyu-Young Whang. 2007. Trajectory Clustering: A Partition-and-Group Framework. In *SIGMOD'07*.
- [17] Wang-Chien Lee and John Krumm. 2011. Trajectory Preprocessing. In *Computing with Spatial Trajectories*.
- [18] Quannan Li, Yu Zheng, Xing Xie, Yukun Chen, Wenyu Liu, and Wei-Ying Ma. 2008. Mining User Similarity Based on Location History. In *SIGSPATIAL'08*.
- [19] Zhenhui Li, Bolin Ding, Jiawei Han, Roland Kays, and Peter Nye. 2010. Mining Periodic Behaviors for Moving Objects. In *SIGKDD'10*.
- [20] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *arXiv preprint arXiv:1301.3781*.
- [21] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.
- [22] Johannes Paefgen, Florian Michahelles, and Thorsten Staake. 2011. GPS trajectory feature extraction for driver risk profiling. In *Proceedings of the 2011 International workshop on Trajectory data mining and analysis*.
- [23] Swaminathan Sankaranarayanan, Pankaj K Agarwal, Thomas Molhave, Jiangwei Pan, and Arnold P Boedihardjo. 2013. Model-driven matching and segmentation of trajectories. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 234–243.
- [24] Renchu Song, Weiwei Sun, Baihua Zheng, and Yu Zheng. 2014. PRESS: A Novel Framework of Trajectory Compression in Road Networks. In *VLDB'14*.
- [25] Greg Corrado, Tomas Mikolov, Kai Chen, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *ICLR Workshop* (2013).
- [26] Michail Vlachos, George Kollios, and Dimitrios Gunopulos. 2002. Discovering Similar Multidimensional Trajectories. In *ICDE'02*.
- [27] Duncan J Watts and Steven H Strogatz. 1998. Collective dynamics of small-world networks. *nature* 393, 6684 (1998), 440.
- [28] Wenming Xiao, Xia Xu, Kang Liang, Junkang Mao, and Jun Wang. 2016. Job recommendation with Hawkes process: an effective solution for RecSys Challenge 2016. In *Proceedings of the Recommender Systems Challenge*. ACM, 11.
- [29] Byoung-Kee Yi, HV Jagadish, and Christos Faloutsos. 1998. Efficient Retrieval of Similar Time Sequences Under Time Warping. In *ICDE'98*.
- [30] Nicholas Jing Yuan, Yu Zheng, Lihuang Zhang, and Xing Xie. 2013. T-Finder: A Recommender System for Finding Passengers and Vacant Taxis. In *TKDE'13*.
- [31] Yu Zheng, Like Liu, Longhao Wang, and Xing Xie. 2008. Learning Transportation Mode from Raw GPS Data for Geographic Applications on the Web. In *WWW'08*.
- [32] Yu Zheng and Xing Xie. 2011. Learning Travel Recommendations from User-generated GPS Traces. In *TIST'11*.
- [33] Yu Zheng, Lizhu Zhang, Xing Xie, and Wei-Ying Ma. 2009. Mining interesting locations and travel sequences from GPS trajectories. In *Proceedings of the 18th international conference on World wide web*. ACM, 791–800.