

Semantic Trajectory Representation and Retrieval via Hierarchical Embedding

Chongming Gao¹ · Chen Huang¹ ·
Zhong Zhang¹ · Qinli Yang¹ · Junming
Shao¹

Received: date / Accepted: date

Abstract Trajectory mining has gained growing attentions due to its wide emerging applications, such as location-based service, urban computing and movement behavior analysis. One critical and fundamental mining task is to retrieve certain locations or trajectories that satisfy particular patterns. However, it becomes a tricky problem to query various semantic patterns from traditional trajectory databases, since most existing approaches mainly represent the trajectory as a collection of geographic and temporal features, and the latent semantic properties are little considered. In this paper, we introduce a new semantic trajectory representation method, incorporating trajectory structures, temporal information as well as domain knowledge to make efficient semantic retrieval possible. Specifically, to extract structures from disordered raw trajectories, a synchronization-based model is first introduced to identify multi-resolution regions of interest (ROIs). Relying on the hierarchical ROI network, a hierarchical embedding model is further proposed to embed ROIs as well as trajectories as continuous vectors. Most importantly, the metric in this embedding vector space is tailored to express the semantic similarity. As a result, users can easily retrieve desirable ROIs or trajectories by computing the Euclidean distance among embedded vectors. Empirical experiments

Chongming Gao
E-mail: chongming.gao@gmail.com

Chen Huang
E-mail: huangc.uestc@gmail.com

Zhong Zhang
E-mail: zhongzhang@std.uestc.edu.cn

Qinli Yang
E-mail: qinli.yang@uestc.edu.cn

Junming Shao
E-mail: junmshao@uestc.edu.cn

¹ School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China

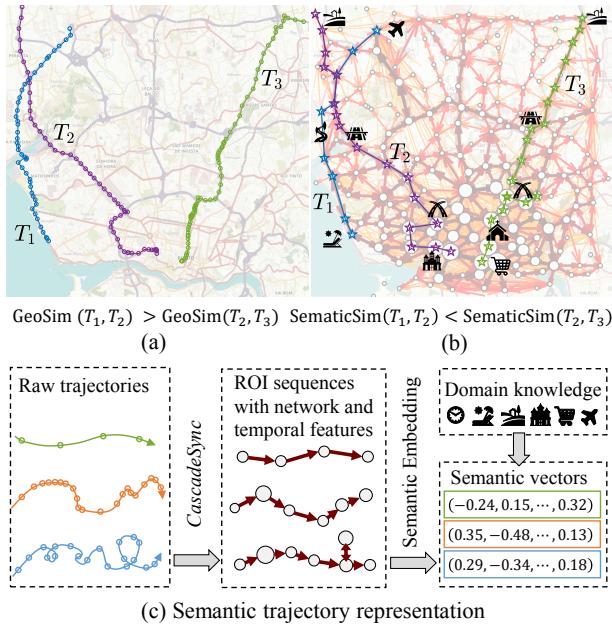


Fig. 1 Illustration of Trajectory representation. (a) Three raw trajectories in Porto, Portugal. (b) The same trajectories represented by nodes in the semantic ROI network. (c) Steps of transformation from raw trajectories to semantic vectors.

on synthetic data and four real-world data sets show that our approach extracts semantic trajectory information effectively, which allows retrieving more similar and interpretable trajectories, by comparing with four state-of-the-art similarity metrics, including DTW, LCSS, EDR and Fréchet distance.

Keywords Trajectory Representation · Clustering · Semantic Embedding · Semantic Retrieval

1 Introduction

The increasing use of location-aware devices has boosted the generation of trajectory data in diverse fields. Generally, a trajectory is represented as a sequence of time-stamped geographic locations, which is used to characterize the movement of objects, such as people, vehicles, animals, hurricanes and ocean currents Mazimpaka and Timpf (2016). Recently, due to the explosion and increasing use of taxi service software, for example DiDi and Uber, and the social media such as Facebook and Foursquare, raw trajectory data have been enriched with various semantic information. Mining these semantic knowledge is a key to understand human mobility, which makes semantic trajectory mining a prevailing subject Alvares et al (2007); Bogorny et al (2009).

To extract knowledge from trajectory data, the primary step is to generate a good trajectory representation. However, it still remains to be a challenging

task, due to the disordered nature of trajectory data, such as different sampling rates and varying lengths of trajectories in real-world applications. To present, depending on related mining jobs, there are mainly three strategies for trajectory representation: point-based Yuan et al (2013); Zheng and Xie (2011); Anagnostopoulos et al (2017), segment-based Bellman (1961); Lee et al (2007); Lee and Krumm (2011) and feature-based representation Annoni and Forster (2012); Ardakani and Hashimoto (2017); Pelekis et al (2017). Based on the representative strategies, a bunch of similarity measures are defined (e.g., DTW Yi et al (1998), LCSS Vlachos et al (2002), EDR Chen et al (2005) and Fréchet distance Eiter and Mannila (1994)).

Unfortunately, the mainstream trajectory representations mentioned above mainly focus on geographic information and barely consider specific semantic knowledge. Thus, the indexing systems established on the representations are limited by few types of retrieval tasks, which spawns a bunch of isolated researches. For example, there is no system designed for multi-faceted queries like: (1) Retrieve some suspected regions in the city that bear the most resemblance to the given crime areas. (2) Retrieve the most suspected movement tracks from trajectory databases by giving criminals' classic course of actions. For such complex retrieval tasks, users have to manually dissect the problem into separated parts, then construct queries respectively.

1.1 Basic Idea

In order to achieve aforementioned semantic region and trajectory retrieval tasks, the similarity metric should be tailored to integrate certain desirable semantic similarity. For illustration, Fig. 1(a) displays three taxi trajectories (i.e., T_1 , T_2 and T_3) in Porto, Portugal. If we only consider the geometric properties of the three trajectories, the similarity between T_1 and T_2 is higher than that between T_2 and T_3 . However, by taking into account the transport system (i.e., structural information) and region functions (i.e., domain knowledge) of city, T_2 is more similar to T_3 (Fig. 1(b)). As both trajectories indicate the movement from rural area to the downtown through the motorways, while T_1 means the movement from airport to the seashore area through the national roads.

In this work, we propose a semantic trajectory representation, by which the trajectory structures are extracted and the desirable semantic information is incorporated. To begin with, a synchronization-based clustering model, CASCADESYNC, is proposed to transform raw GPS points to multi-resolution regions of interest (ROIs) from fine-grained to coarse-grained levels. Every ROI in the tree-shaped hierarchy, with a certain representative range, is a compact prototype of a large set of GPS points or small ROIs. Therefore, trajectories can be replaced by sequences of ROIs on each hierarchical level, and the multi-resolution ROI hierarchy is further used to generate a hierarchical ROI network (Fig. 4(a)). Afterwards, by leveraging network structures, temporal information and other domain knowledge, a hierarchical embedding model is

proposed to embed every ROI/trajectory to a continuous vector. In this way, the semantic information is well-preserved in the metric of the embedding space. The whole procedure is shown in Fig. 1(c). Utilizing the representation, the semantic retrieval task can be cast as the problem of searching the most similar vectors to the target vector by simply computing the Euclidean distance between two vectors.

1.2 Main Contributions

The main contributions of this work are summarized as follows.

1. We propose a synchronization-based clustering algorithm CASCadesYNC to extract trajectory structures, representing as a multi-resolution ROI network. The geographic and temporal information as well as global statistics are well-preserved on the hierarchical ROI network.
2. Utilizing the geometric property and semantic information (network structures, temporal information and domain knowledge), we propose a hierarchical embedding model to embed each ROI/trajectory as a continuous vector in a semantic vector space. Thereby, the semantic similarity between two ROIs/trajectories can be measured by computing the Euclidean distance of two vectors directly.
3. The empirical experiments show the proposed representation successfully captures semantic information of trajectory data, and renders better performance than the state-of-the-art metrics in semantic ROI/trajectory retrieval tasks.

The rest of paper is organized as follows. Section 2 discusses related work. Section 3 elaborates the proposed method. Section 4 presents the experiments. Finally, we conclude our work in Section 5.

2 Related Work

2.1 Trajectory Representation and Retrieval

Trajectory representation, indexing and retrieval are the foundations of trajectory pattern mining. Here we introduce a brief summary of state-of-the-art trajectory representation and retrieval strategies.

2.1.1 Trajectory Representation

Currently, there are mainly three strategies to represent trajectory: point-based representation, segment-based representation and feature-based based representation. Point-based representation is the most prevalent method. The key idea is to detect the most representative points and use them to describe

trajectories. To define the representative point, Some researchers use the concept of *stay points*, which are the small regions where the moving objects stop for relative long time Li et al (2008); Zheng and Xie (2011). Li et al (2010) detect a few of *reference spots* where a moving object has visited frequently.

Segment-based representation is also popular. Lee et al (2007) partition trajectories to segments by using the concept of Minimal Description Language (MDL). Zheng et al (2008b) divide trajectories into alternate *walk segments* and *nonwalk segments* based on locations and velocities of moving objects. Moreover, some work matches trajectories to road map to get restrained segments Kellaris et al (2009); Song et al (2014).

Feature-based representation aims to construct a new feature space to represent trajectories. Annoni and Forster (2012) transfer trajectories to Fourier coefficients, simplifying a 2D motion representation to a 1D complex sequence representation. Gariel et al (2011) resample trajectories to have a unified length, then use principal component analysis (PCA) to pursue a compact representation. And some work Ardakani and Hashimoto (2017); Gao et al (2017) feeds trajectories to Long-Short Term Memory (LSTM) networks to learn the spatial path features encoded in hidden vectors.

2.1.2 Trajectory Retrieval

Usually, the queries of trajectories are heuristic: Ask for points of interest (POIs) or trajectories that bear the most resemblance to a specified POI/trajectory. The most critical task is to define the similarity metric. Almost all mainstream metrics only focus on the geographic similarity. Early researchers Agrawal et al (1993) use the sum-of-pairs distance, requiring trajectories to have a unified length, which is not realistic in real-world data. Dynamic time warping (DTW) distance is the first one to overcome the defect. It allows some points repeating multiple times in order to obtain optimal alignment Yi et al (1998); Shokoohi-Yekta et al (2017). The longest common subsequence (LCSS) distance Vlachos et al (2002) and edit distance on real sequence (EDR) distance Chen et al (2005) are proposed to get rid of the effects causing by the noise points. Fréchet distance is another novel similarity measure between curves taking into account the locations and ordering of the points along the curves Eiter and Mannila (1994).

Building upon the representations and measures, many trajectory indexing structures are proposed. For examples, STR-tree Pfoser et al (2000), TB-tree Pfoser et al (2000) and HR-tree Nascimento and Silva (1998) generalize the R-tree, an effective spatial database, to store spatial and temporal dimensions together. Afterwards, SETI Chakka et al (2003) is proposed to distinguish temporal information from spacial indexing system to increase the retrieval efficiency.

However, almost all traditional trajectory representation models and indexing systems are built upon the geographic and temporal features of trajectory data, thus the semantic retrieval and mining tasks are hard to perform.

2.2 Distributed Vector Embedding

Word embedding becomes ubiquitous in natural language processing (NLP) and information retrieval (IR). It captures the syntactic and semantic similarities among words, and exhibits the effectiveness on large data sets Bengio et al (2003); Collobert et al (2011); Mikolov et al (2013); Pennington et al (2014). The philosophy of word embedding is to project words to a continuous vector space in an unsupervised way, by exploiting the company relationship of words in documents. Afterwards, the semantic meanings can be computed by the mathematics of embedding vectors. Recently, a large body of works have tried to compute the embedding that captures the semantics of word sequences (phrases, sentences, and paragraphs), with methods ranging from simple additional composition of the word vectors Mitchell and Lapata (2010); Iyyer et al (2015) to sophisticated architectures such as recursive neural networks Socher et al (2011), convolutional neural networks Kalchbrenner et al (2014) and recurrent neural networks Tai et al (2015).

Intuitively, ROIs are analogous to words while trajectories are analogous to sentences. However, the similarity metrics of trajectory data are in marked contrast to those of languages. For example, if two words are similar in language semantics, they will be contexts of each other in some sentences. In contrast, two similar ROIs will never adjoin each other in any trajectory, when the geographic distance between them is too large. Therefore, the semantic contexts (neighborhoods) have to be meticulously chosen.

2.3 Synchronization-based Clustering

Our work is also highly related to synchronization-based data mining Böhm et al (2010); Shao et al (2013). Different from traditional clustering algorithms, synchronization-based clustering makes each data point interact with surrounding neighbors, and changes all points' positions dynamically. The model converges when points stop changing positions. Recently, many synchronization-based models and data mining algorithms have been proposed and shown many desirable properties in a wide range of applications Kim et al (2008); Seliger et al (2002); Shao et al (2015, 2017). For instances, in network mining, each node on network is pushed or pulled through the local network structure, resulting in a few of distinct communities Shao et al (2015), Seliger et al (2002) discuss mechanisms of learning and plasticity in networks of phase oscillators through a generalized Kuramoto model. For bioinformatics, a strategy is proposed to find groups of genes by analyzing the cell cycle specific gene expression Kim et al (2008). From another perspective, the correlated genes and conditions can interact simultaneously, and a set of co-clusters are yielded Shao et al (2017). The concept of synchronization is suitable for trajectory data, since it does not need to determine k explicitly like k-means based models, either does it render clusters in arbitrary shapes like DBSCAN based models.

Table 1 Important notations

Notation	Description
l	$l = \{0, 1, 2, \dots, L_M\}$ is layer of ROI networks.
\mathcal{T}^l	Trajectory set, each $T_i^l \in \mathcal{T}^l$ is a trajectory written as (2).
\mathcal{ROT}^l	ROI set, each $ROI_i^l \in \mathcal{ROT}^l$ is a node of network on the l -th layer ($l > 0$).
\mathcal{P}^l	Point set, each $P_i^l \in \mathcal{P}^l$ is a GPS point ($l = 0$) or center of ROI_i^l ($l > 0$).
ϵ	The radius of interaction range, defined in (3).

3 Proposed Method

Notations. The most prevalent data format of the trajectory is a temporal sequence of latitude/longitude coordinates as follows:

$$T = \{\langle s_1, s_2, \dots, s_n \rangle | s_i = (p_i, t_i)\}, \quad (1)$$

where $p_i = (x_i, y_i)$ is a (*latitude, longitude*) pair (GPS coordinate), representing a sample point on the map. t_i is the time stamp of sample s_i , and n is the length of this trajectory.

In our work, as mentioned above, a massive amount of GPS points are reduced to a few of multi-resolution regions of interest (ROIs) (Fig. 4(a)). Consequently, the trajectories are reduced to sequences of ROIs on different layers of the hierarchical ROI network. The trajectory represented by ROIs is written as,

$$T^l = \{\langle s_1^l, s_2^l, \dots, s_m^l \rangle | s_i^l = (p_i^l, t_i^l)\}, \quad (2)$$

where the superscript l denotes the layer of hierarchical ROI network, p_i^l is the center point of the ROI that replaces one or more raw GPS points on raw trajectory T . In addition, some key notations used in this work are listed in Table 1.

3.1 Multi-resolution Regions of Interest Extraction via CascadeSync Model

As trajectory data is represented by a mass of GPS points. It is not a good way to extract patterns and mine the trajectory data on these points directly, as different trajectories often have varying lengths and different sampling rates. One intuitive way is to reduce those raw points to fewer representative prototypes. Here, we employ a synchronization-based clustering to obtain these prototypes.

Typically, a synchronization-based clustering algorithm needs three definitions to simulate a dynamic clustering process: First, a parameter ϵ specifying the interaction range among objects, second, the interaction model for clustering, and finally, a stopping criterion to terminate dynamic clustering. Our approach follows and extends the synchronization-based clustering algorithm, SYNC, which is presented and discussed in full detail in Böhm et al (2010).

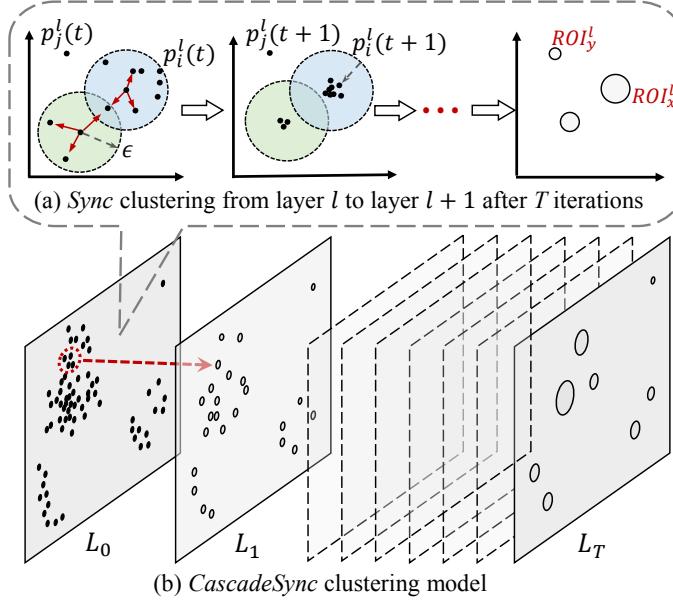


Fig. 2 Illustration of Sync algorithm and CascadeSync algorithm

Definition 1 (ϵ -Range Neighborhood) Given a GPS data set $\mathcal{P} \subset \mathbb{R}^n$, the ϵ -range neighborhood of a GPS point $p \in \mathcal{P}$, denoted as $N_\epsilon(p)$, is defined as:

$$N_\epsilon(p) = \{q \mid dist(p, q) \leq \epsilon\}, \quad (3)$$

where $dist(p, q)$ is a metric distance function. The Euclidean distance is used in this study.

Definition 2 (Interaction Model) Let p be a GPS point on the map. With an ϵ -range neighborhood interaction, the dynamics of the value of the point p is defined as:

$$p(t+1) = p(t) + \frac{1}{|N_\epsilon(p)|} \cdot \sum_{q \in N_\epsilon(p)} \sin(q(t) - p(t)), \quad (4)$$

where $\sin(x)$ is the coupling function, applying to every dimension of vector x . $p(t+1)$ is the renewal position of $p(t)$ during the dynamic clustering, $t \in \{0, \dots, T\}$ denotes the iteration step. Note all dimensions are normalized to $[0, \pi/2]$ to let $\sin(\cdot)$ make sense.

Definition 3 (Cluster Order Parameter) The cluster order parameter r is used to terminate the dynamic clustering by investigating the degree of local synchronization, which is defined as:

$$r(t) = \frac{1}{N} \sum_{i=1}^N \frac{1}{|N_\epsilon(p(t))|} \sum_{q \in N_\epsilon(p)} e^{-||q(t) - p(t)||}, \quad (5)$$

The dynamic clustering terminates when $r(t)$ converges to 1.0, which indicates local phase coherence. At this moment, all cluster points have the same location.

For synchronization-based dynamic clustering, each point is viewed as a phase oscillator and has its own phase (position) at the beginning. As time evolves, each point interacts with its ϵ -range neighborhood according to the interaction model (4). Finally, all locally similar points are synchronized together and prototypes are formed. A prototype also represents the center of a region of interest (ROI). We illustrate this process in Fig. 2(a), where p_i^l is a point that would be absorbed into ROI_x^l with other points in this region. Meanwhile, p_j^l has no neighbors to interact with. It would, consequently, constitute a solitary region ROI_y^l , which aptly preserves information of original data. The salient feature of synchronization-based clustering is its dynamic property. The position value of point changes in a non-linear way driven by the local data structure. Hence, the derived ROI well preserves the original data structure.

Besides, we extend SYNC to a hierarchical version, which called CASCADESYNC. The basic idea is quite intuitive. For the first step, we cluster all GPS points with a small interaction range, which usually results in a large number of small clusters (ROIs). Since points in a same cluster have synchronized together, we can use the synchronized points, i.e., centers of ROIs, to characterize all cluster objects. Therefore, a new data set including all these synchronized points (ROIs) are generated. Consequently, we can conduct clustering again with a larger interaction range, and repeat the procedure, as illustrated in Fig. 2(b). However, for a new data set, since each ROI in the new data set characterizes different number of points in the previous layer of data, the interaction model should consider the weight of each point, which is:

$$p^l(t+1) = p^l(t) + \frac{1}{\sum_{q^l \in N_\epsilon(p^l)} w_{q^l}} \cdot \sum_{q^l \in N_\epsilon(p^l)} w_{q^l} \sin(q^l(t) - p^l(t)), \quad (6)$$

where w_{q^l} is the number of points that are represented by the prototype q^l on the layer $l - 1$. The whole algorithm is illustrated in Algorithm 1.

Finally, bridged by trajectories on every layer, all ROIs compose a hierarchical network. For illustration, Fig. 4(a) gives a toy example. In fact, CASCADESYNC not only supports a hierarchical data representation, the speedup of synchronization process is another desirable property. The reason is with smaller interaction range, CASCADESYNC is easier to converge. Although more clusters are generated, they are viewed as new objects, and can be further clustered efficiently.

3.2 Learning Semantic Trajectory Representation via Hierarchical ROI Embedding Model

After extracting the hierarchical ROI network, each trajectory is represented as sequences of ROIs on each layer of the network. Although such structure

Algorithm 1: Hierarchical ROI Extracting via CASCADESYNC

```

Input      : Trajectory dataset  $\mathcal{T}$ ;
Output     : Hierarchical ROI network  $G(V, E)$ ;
Parameter: Interaction range  $\epsilon_0$ ; Incremental change  $\Delta\epsilon$ ;
1  $\mathcal{P}^0$  = Raw GPS point set of  $\mathcal{T}$ ;
2  $\mathcal{P}^0 = \text{Norm}(\mathcal{P}^0)$ ;                                 $\triangleright$  Normalized each dimension to  $[0, \pi/2]$ .
3 Layer  $l = 0$ ; Interaction range  $\epsilon = \epsilon_0$ ;
4 while layer  $l \leq L_M$  do
5    $t = 0, \mathcal{P}_t^l = \mathcal{P}^l, SyncLoopFlag = \text{TRUE}$ ;
6   while  $SyncLoopFlag == \text{TRUE}$  do
7     foreach point  $p_i^l(t) \in \mathcal{P}_t^l$  do
8       | Search its  $\epsilon$ -range neighbors  $N_\epsilon(p_i^l(t))$  with Eq.(3);
9       | foreach neighbors  $q_j^l(t) \in N_\epsilon(p_i^l(t))$  do
10      |   | Compute  $p_i^l(t+1)$  with Eq.(6);
11      |   end
12    |   end
13    |   Compute order parameter  $r(t)$  with Eq.(5);
14    |    $t = t + 1; \epsilon = \epsilon + \Delta\epsilon$ ;
15    |   if  $r(t)$  converges then  $SyncLoopFlag = \text{FALSE}$ ;
16  end
17   $\mathcal{ROI}^l = \{p_1^l(t), \dots, p_N^l(t)\}$ ;
18   $\mathcal{P}^{l+1} = \mathcal{ROI}^l, l = l + 1$ ;
19 end
20 Build a hierarchical network  $G(V, E)$  by using  $\{\mathcal{ROI}^1, \dots, \mathcal{ROI}^{L_M}\}$ ;

```

provides a more compact way to characterize the trajectory, the semantic information of the whole trajectory is not well exploited. Inspired by word embedding scheme in natural language processing (NLP), we propose an embedding model to further transform ROIs and trajectories to continuous vectors in a semantic space, so as to uncover the hidden semantic information in trajectory data.

3.2.1 Semantic ROI Embedding on One-layer Network

The key of embedding model is to select the neighborhoods, or contexts, of a given object so that the relationships among objects could be investigated. Therefore, we consider the embedding neighborhoods of a ROI from two aspects: geometric neighborhood and semantic neighborhood.

Geometric Neighborhood. Given a ROI, its geometric neighbors are those surrounding ROIs which can transport to or be transported from the target ROI in W -hops on the ROI network. Fig. 3 plots the geometric neighbors with $W = 2$, which are indicated by green and blue circles. For an example, ROI_k is one of the geometric neighborhood of ROI_i .

Semantic Neighborhood. As a ROI is not just a node on network, it also represents a region enriched with various semantic information. To encode the rich semantics, we define the semantic neighborhoods of a given ROI from following perspectives:

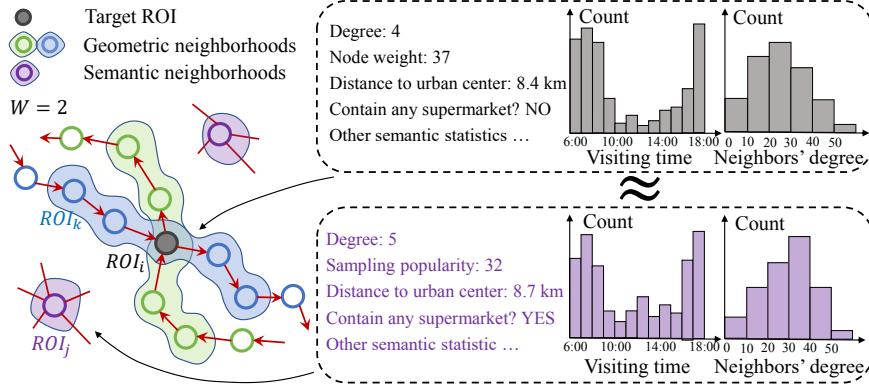


Fig. 3 Illustration of embedding neighborhoods (contexts) of a given target ROI.

- *Network properties.* The hierarchical ROI network is a compact representation of trajectory data, from which the general statistics of transportation structure and traffic condition can be extracted. Specifically, the *node weight* of a ROI, i.e., the number of points the ROI represents, can reflect the population density of this region. The *edge weight* measures the strength of transportation flow between two regions. The *degree* of a ROI reflects whether a region is a hub in the city. And *neighbor's degree distribution* of a ROI can reflect, from a higher level, the importance and continuity of the region. For example, if two ROIs are similar to each other in above aspects, maybe they represent a hub bus station and a center railway station, or a huge company district and a university campus respectively.
- *Temporal contexts.* Temporal information is also preserved in the ROI network after applying CASCADESYNC algorithm. The distribution of *visiting time* and *staying duration* can reflect the temporal patterns of a region. For example, a region contains many restaurants can be visited and stayed mainly at lunchtime and dinner time. The regions representing residential communities have the distinct characteristics that most people stay there during the whole night.
- *Domain Knowledge.* Different from the network properties and temporal information, domain knowledge cannot be extract from the trajectory data. Instead, domain knowledge is the auxiliary information incorporating into embedding model as a kind of supervised label. For instance, the distance between one ROI and the center point of city, or whether one ROI represents a road junction, a supermarket, or a dangerous crime area.

An example is illustrated in Fig. 3, where the ROI_j drawn in purple circle is a semantic neighborhood of target ROI_i . Notwithstanding the fact that they are not connected by any trajectory with $W = 2$ hops, the semantics of ROI_j are more similar to those of ROI_i than all other ROI, which results in a strong correlation between them.

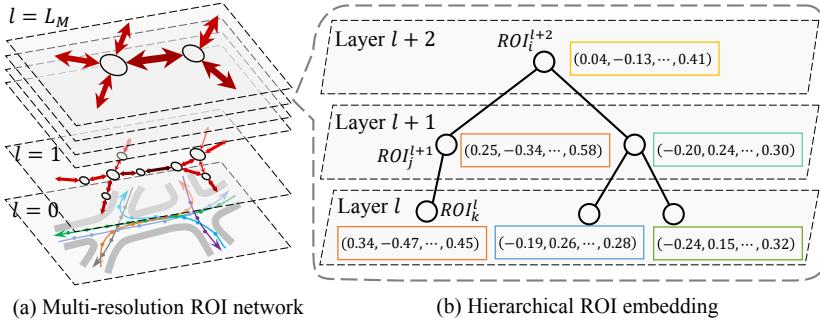


Fig. 4 Illustration of hierarchical ROI embedding on a multi-resolution ROI network

Once the geometric and semantic neighborhoods are obtained, we can embed ROIs into a high dimensional space, borrowing the idea of skip-gram model with negative sampling (SGNS) architecture Mikolov et al (2013).

Embedding Model. Given a trajectory data set \mathcal{T} , for any layer on the generated hierarchical ROI network, let w be any ROI on this layer, $N(w)$ be the collection of all geometric and semantic neighborhood ROIs of w , and $NEG(w)$ be the negative sampling set of w , which can be drawn using the distribution of network nodes' weight. We define the probability of transportation from a source ROI $u \in N(w)$ to the target ROI v as:

$$p(v|u) = \begin{cases} \sigma(\mathbf{t}_w \cdot \mathbf{s}_u) & \text{if } v = w, \\ 1 - \sigma(\mathbf{t}_v \cdot \mathbf{s}_u) & \text{otherwise.} \end{cases} \quad (7)$$

where $\sigma(x) = 1/(1 + \exp(-x))$ is the sigmoid function. \mathbf{s}_u and \mathbf{t}_v are the source and target vectors of ROI u and v respectively.

Accordingly, the likelihood function is written as follows.

$$\begin{aligned} L_{H^l} &= \log \prod_{w \in \mathcal{ROI}^l} \prod_{u \in N(w)} \sigma(\mathbf{t}_w \cdot \mathbf{s}_u) \prod_{v \in NEG(w)} [1 - \sigma(\mathbf{t}_v \cdot \mathbf{s}_u)] \\ &= \sum_{w \in \mathcal{ROI}^l} \sum_{u \in N(w)} \left\{ \log \sigma(\mathbf{t}_w \cdot \mathbf{s}_u) + \sum_{v \in NEG(w)} \log [1 - \sigma(\mathbf{t}_v \cdot \mathbf{s}_u)] \right\}, \end{aligned} \quad (8)$$

The likelihood function indicates in the embedding space, the vector of target ROI should locate as close as possible to all vectors of its neighborhoods, while distinguishing from the vectors of its negative samples.

3.2.2 Feature Propagation through the Multi-layer Networks

The embedding process is conducted on every layer of the hierarchical ROI network. Moreover, considering the fact every ROI has parent or children (unless it is on the top or bottom layer), we make an intuitive assumption: Every ROI should be embedded as close as its parent ROI (if exists). The assumption

is justified under urban semantics. For example, the semantic meanings of a business center should contain all the functions and utilities that are provided by the companies or shops located in the region. Therefore, The embedding model is augmented with a regularization term:

$$L_{V^l} = \sum_{w \in \mathcal{ROT}^l} \frac{1}{2} \|\mathbf{t}_w - \mathbf{t}_{\pi(w)}\|_2^2 + \frac{1}{2} \|\mathbf{s}_w - \mathbf{s}_{\pi(w)}\|_2^2, \quad (9)$$

where $\pi(w)$ is its parent of ROI w . Actually, this term regulates the embedding vectors by propagating influences from parent nodes to children nodes on the hierarchical ROI network. An example is shown in Fig. 4(b), where the ROI_j^{l+1} is a node on layer $l+1$ of hierarchical ROI network, and the vector of ROI_j^{l+1} is constrained by vectors of its parent node ROI_i^{l+2} and its child node ROI_k^l .

3.2.3 Hierarchical ROI Embedding on Multi-layer Networks

Eventually, by integrating embedding model (8) and regularization term (9), the objective function on the whole hierarchical ROI network is reformulated as follows.

$$\max \sum_{l=1:L_M} L_{H^l} - \alpha \sum_{l=1:L_M-1} L_{V^l} \quad (10)$$

where α is a trade-off coefficient. For the optimization of embedding model (10), we apply the stochastic gradient descent (SGD) algorithm. The derivatives of parameters are easily computed and updated as lines 7-17 in Algorithm 2.

3.2.4 Trajectory Semantic Embedding.

Once we derive the semantic embedding for each ROI, the semantic vector of a trajectory can be easily obtained by summing up all feature vectors of corresponding ROIs with weights. Formally, given a trajectory $T^l = \{ROI_1^l, ROI_2^l, \dots, ROI_n^l\}$ on layer l , the embedding vector, denoted as $\text{vec}(T)$, is defined as:

$$\text{vec}(T) = \sum_{l=1}^{L_M} \sum_{i=1}^n \tau_i^l \cdot \text{vec}(ROI_i^l), \quad (11)$$

where τ_i^l is the weight of ROI_i^l , defined as the time duration of trajectory T staying at ROI_i^l . Here, some sophisticated strategies (e.g., a time decay function) can be used to capture detailed temporal information.

It may seem a little unsophisticated at the first glance, comparing to those deep model tailored for sentence embedding Socher et al (2011); Kalchbrenner et al (2014); Tai et al (2015). However, the addition model is prevalent and shown to be very effective for sentence embedding Blacoe and Lapata (2012). Wieting et al (2016) even show the model, which embeds the sentences by simply averaging the vectors of words, beats almost all neural network based models. Besides, it also agrees to the common sense: A trajectory can be completely characterized by its element ROIs with order.

Algorithm 2: Semantic Trajectory Embedding

Input : Hierarchical ROI network $G(V, E)$; Geometric and semantic neighborhoods $N(w)$;
Output : Semantic vectors: $\text{vec}(ROI)$ and $\text{vec}(T)$;
Parameter: Times of iteration K , Learning rate η ; Trade-off coefficient α ;

```

1 foreach  $ROI_i^l \in \mathcal{ROT}^l$  do RandomlyInit ( $s_i, t_i$ );
2 foreach iteration  $k \in \{1, 2, \dots, K\}$  do
3   foreach layer  $l \in \{1, 2, \dots, L_M - 1\}$  do
4     foreach ROI  $w \in \mathcal{ROT}^l$  on the ROI network do
5       Search geometric and semantic neighborhood  $N(w)$ ;
6       Sample its negative samples  $NEG(w)$ ;
7       foreach  $u \in N(w)$  do
8          $e = 0$ ;
9         foreach  $v \in \{w\} \cup NEG(w)$  do
10          if  $v == w$  then  $I(v) = 1$ ; else  $I(v) = 0$ ;
11           $g = \eta(I(v) - \sigma(t_v \cdot s_u))$ ;  $e = e + gt_v$ ;
12           $t_v = t_v + gs_u + \eta\alpha(t_{\pi(v)} - t_v)$ ;
13           $t_{\pi(v)} = t_{\pi(v)} + \eta\alpha(t_v - t_{\pi(v)})$ ;
14        end
15         $s_u = s_u + e + \eta\alpha(s_{\pi(u)} - s_u)$ ;
16         $s_{\pi(u)} = s_{\pi(u)} + \eta\alpha(s_u - s_{\pi(u)})$ ;
17      end
18    end
19  end
20 end
21 foreach  $ROI_i \in \mathcal{ROT}$  do  $\text{vec}(ROI_i) = [s_i, t_i]$  ;
22 foreach trajectory  $T \in \mathcal{T}$  do Compute  $\text{vec}(T)$  via Eq.(11);

```

3.3 Semantic ROI and Trajectory Retrieval

In reality, many patterns are contained implicitly in trajectories, which means the queries cannot be conducted directly. As an alternative, we can query the most similar places or trajectories of a selected place or trajectory. Namely, we can retrieve all ROIs/trajectories ordered by the similarity to a certain sample, rather than constructing complex queries directly.

Since ROIs/trajectories are represented as vectors in a continuous space with semantics, we can retrieve the most similar ROIs/trajectories by computing the distances among the embedding vectors. Without loss of generality, we use Euclidean distance in this work.

3.4 Time Complexity Analysis

Our semantic trajectory representation mainly includes two parts: ROI extraction and semantic embedding. For ROI extraction via CASCADESYNC model, the time complexity is $O(L \times T \times (N_l \times \log(N_l)))$, where N_l is the number of points, which reduced exponentially over the layer l , T is the time steps in each round and L is number of layers in CascadeSync. Usually, L is small with $L \leq 20$ in practice. The time complexity of ROI embedding is

$O(K \times N \times N_1 \times N_2)$, where N is the number of points, N_1 is the number of geometric and semantic neighborhoods, N_2 is number of negative samples, and K is the number of iterations.

4 Experiments

4.1 Experimental Setup

4.1.1 Data Sets

We evaluate the proposed method on synthetic data and four real-world trajectory data sets.

Synthetic Data. To prove the concepts, we generate 50 random trajectories, given the random start point and end point, using a probabilistic path planning algorithm Kavraki et al (1996). To define the semantics in the trajectories, we randomly select 10 points as *key points*, and ensure each key point is passed by at least one trajectory. The semantic ROI and trajectory retrieval tasks are aimed to detect the ROIs and trajectories highly related to these key points.

Real-World Data. We evaluate our proposed method on four trajectory data: Geolife Zheng et al (2008a, 2009, 2010), Kaggle Taxi, T-Drive Yuan et al (2010, 2011) and Chengdu. The statistics of four data sets are listed in Table 2.

4.1.2 Evaluation Metrics

In order to measure the accuracy of semantic retrieval results, we need to define the semantic ground-truth of ROI and trajectory. In the synthetic experiment, the semantic ground-truth is implied in the 10 key points. For similar ROI retrieval task, given any key point, the most similar ROI should be those ROIs that contain other nine key points, which are referred as ground-truth ROIs, or key ROIs. The retrieval performance is measured by the ratio of correctly detected those true ROIs. For trajectory retrieval task, we select the target trajectory as one that passes ground-truth ROIs as more times as possible. The most similar trajectories of the target, in this semantics, should contain as more ground-truth ROIs as possible. Therefore, we can obtain the

Table 2 Information of four real-world data sets

Dataset	#Point	#Traj.	[Longitude, Latitude range]	[Width×Height](m)
Geolife	24,876,978	18,670	[116.194, 116.553, 39.751, 40.033]	[31,024 × 31,368]
T-Drive	6,969,481	8,768	[116.194, 116.553, 39.751, 40.033]	[31,024 × 31,368]
Kaggle	78,239,735	1,704,769	[-8.702, -8.549, 41.135, 41.246]	[12,613 × 12,365]
Chengdu	9,671,104	2,003	[103.913, 104.180, 30.538, 30.765]	[25,484 × 25,219]

ground-truth order of retrieved trajectories, and the performance can be measured by comparing the ground-truth order and retrieved order via computing the normalized discounted cumulative gain at a particular rank position k (NDCG@ k), which is defined as below.

$$\text{NDCG}@k = \frac{\text{DCG}@k}{\text{IDCG}@k}, \quad (12)$$

where IDCG@ k is the *ideal* DCG@ k value of the ground-truth order. And DCG@ k is defined as:

$$\text{DCG}@k = \sum_{i=1}^k \frac{2^{rel_i} - 1}{\log_2(i + 1)}, \quad (13)$$

where rel_i is the ordinal relevance of the result at position i . For clarity, we use 5 levels of relevance throughout all experiments, and ensure each level contains objects equally.

Real-world Region Ground-truth Acquisition. Real-world trajectory data are enriched with various semantic information, however, the ground-truth labels are usually hard to obtain. Fortunately, benefited from reverse geocoding function of online map services, MapBox API¹ and Amap API², we can obtain information of regions, which can serve as the domain knowledge as well as the ground-truth labels in the experiments. Specifically, the APIs can convert GPS coordinates to function categories, such as bank, school, or market. Therefore, We define the semantic ground-truth label of every ROI as its distribution of function categories, and the ground-truth similarity of two ROIs A and B is defined as the common area of two distributions:

$$\text{Sim}(A, B) = \sum \min(\text{distribution}(A), \text{distribution}(B)). \quad (14)$$

For example, the category distributions of two places are A: {bank, school, market, music} = {0.2, 0.7, 0.1, 0}, B: {bank, school, market, music} = {0.4, 0.3, 0, 0.3}. Then $\text{Sim}(A, B) = 0.2 + 0.3 + 0 + 0 = 0.5$. By the way, the reason we use (14) rather than other measures, e.g. KL-divergence, is the fact it is intuitive and suitable for sparse distribution measuring.

Real-world Trajectory Ground-truth Acquisition. The semantic ground-truth labels and ground-truth similarity of ROIs are easily obtained by utilizing the map APIs and (14) respectively. The difficulty lies in obtaining the semantic ground-truth of trajectories. Here, we use an intuitive method: A trajectory is composed of a set of sampling points, so the ground-truth label of this trajectory can be computed by the normalized sum of the ground-truth labels of its sampling points. Besides, it is too trivial and expensive to use raw GPS points to call the map APIs. So we use CASCadesync to derive a single-layer ROI network with a very small interaction range ($\epsilon = 1/200 * (w + h)$,

¹ <https://www.mapbox.com/>

² <https://www.amap.com/>

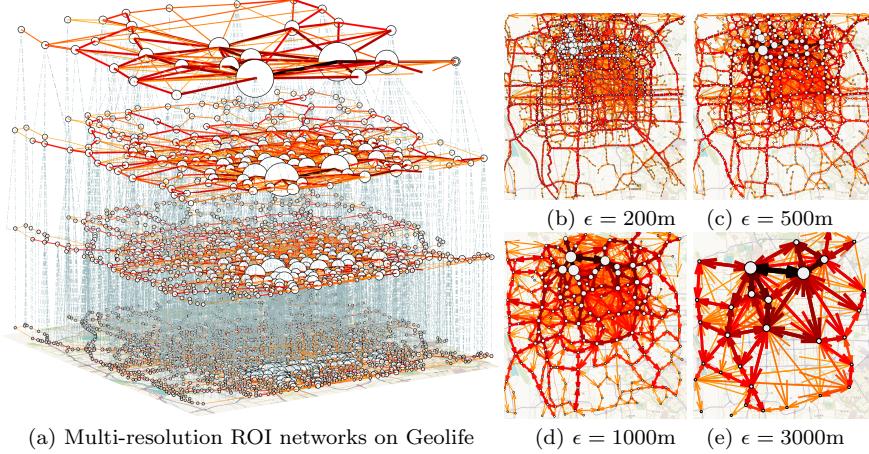


Fig. 5 Multi-resolution ROI networks on Geolife data with four layers. The sizes of ROIs are proportional to the number of represented points. The edge width and color shade are proportional to the transports flow among the ROIs.

where w, h are width and height of map of data set) to reduce the millions of GPS points to thousands of ROIs. Then the ground-truth labels and similarity (using (14)) of trajectories can be easily obtained. Note later we will use this auxiliary ROI network.

Afterwards, we can perform the semantic ROI/trajecotry retrieval, and compute the NDCG@k between retrieved ROIs/trajecotries and ground-truth. Moreover, we can compare the performance with state-of-the-art similarity metrics: DTW, LCSS, EDR and Fréchet distance. Without loss of generality, the four comparison metrics are computed on the auxiliary ROI network mentioned above.

Trivial Solution Removal. For semantic trajectory retrieval, if retrieved trajectories overlap with querying trajectory, i.e. they share most ROIs with querying trajectory, the NDCG@k value will be perfect. However, this trivial solution reflects no semantic similarity. Thereby, we stipulate that a retrieved trajectory is valid if and only if it shares less than 50% of ROIs with querying trajectory on the auxiliary ROI network.

4.2 Proof of Concept

We start with the experiments on synthetic data to prove the basic properties of our proposed approach. We randomly generate trajectories and key points in a box with side length being 100m, illustrated in Fig. 6(a). With varying interaction ranges $\epsilon = \{2, 4, 6, 8, 10\}(m)$, we apply CASCadesync model to generate a 5-layer hierarchical ROI network. The bottom layer ROI network is visualized in Fig. 6(b). The key points, which are plotted as grey pentagons on the map, represent important places in reality, e.g. fire stations, crime area. In

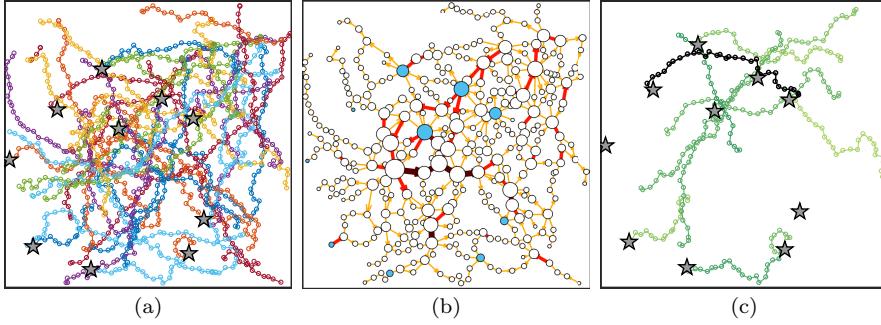


Fig. 6 Illustration of synthetic data and evaluation results. (a) Raw data with key points. The 10 pentagons are randomly generated key points. (b) The bottom layer of the hierarchical ROI network. The blue ROIs are the key ROIs contain at least one key point. (c) Trajectory retrieval result. The black trajectory is the target. The green trajectories in different shades are retrieved results. The deeper the color, the more similar to the target, i.e. containing more key ROIs.

the following, we design two experiments of semantic evaluation and trajectory retrieval.

4.2.1 Key Point Detection on Synthetic Data

To examine whether our representation allows correctly embedding information of key points or not, we perform an experiment by retrieving the most similar region of a given key point. In this task, we use the domain knowledge: whether a ROI contains any key point. To test whether those key ROIs have similar representations, we evaluate the retrieval accuracy of the most similar

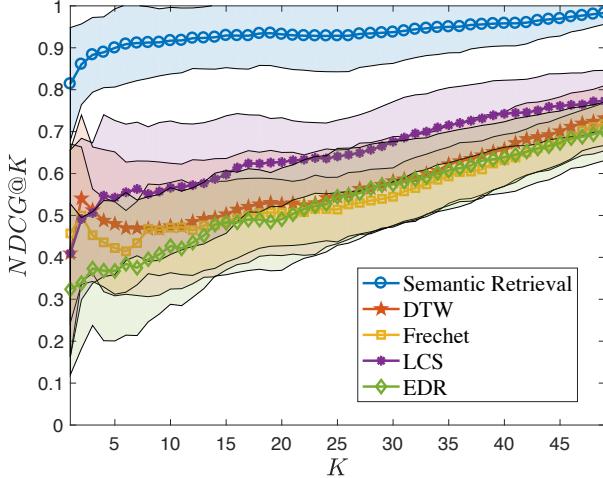


Fig. 7 Semantic trajectory retrieval on synthetic data. The shadow regions represent the bound of one standard deviation of 100 experiments on independent random data sets.

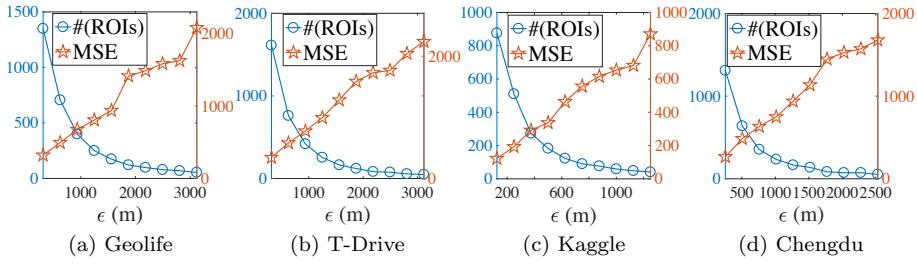


Fig. 8 Exploring ϵ 's effects on four real-world data sets.

ROIs for a given key ROI (blue ROI in Fig 6 (b)). The ideal retrieval result is that all key ROIs are more similar than those non-key ROIs.

Building upon the embedding vectors of ROIs and trajectories, we search the most similar POIs. The experiment is repeated 100 times on random generated data sets independently. The resulting average accuracy is **99.64%**, demonstrating the semantics of key points are successfully incorporated into the embedding representation, and the key points can be detected effectively.

4.2.2 Semantic Trajectory Retrieval on Synthetic Data

Here, we conduct similar experiment on the trajectory level. The query is a target trajectory that contain many key ROIs. Thus, the similar trajectories should contain as many key ROIs as possible. We compute and rank the similarity of rest 49 trajectories to the target, and measure the quality of ranking using NDCG@k in (12). One result is visualized in Fig. 6(c), in which the color shade of green trajectories is proportional to retrieval order. Moreover, we conduct the same experiment using four state-of-the-art metrics on the first layer ROI network (the finest level that has the key ROI labels). The result is shown in Fig. 7. The experiment are repeated 100 times, and the average NDCG@k value as well as the standard deviation are given. From the figure, we can observe that our model beats other four models, and the semantic information is well captured.

4.3 Evaluation on Real-World Data

4.3.1 Effect of Key Parameter

The interaction range ϵ , which controls the trade-off of compression degree and representative accuracy, is the most critical parameter in our model. In order to measure the information loss with different values of interaction range ϵ , we use the Mean of Squared Error (MSE), defined as:

$$\text{MSE} = \frac{1}{|\mathcal{P}|} \sum_{i=1}^{|\mathcal{P}|} (P_i - ROI_j)^2, \quad (15)$$

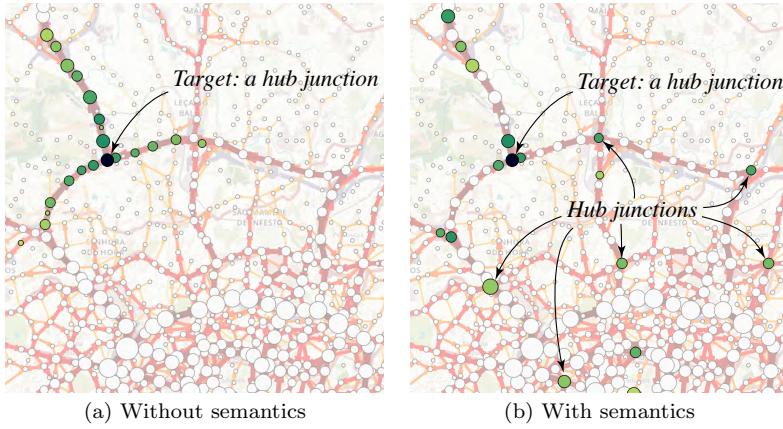


Fig. 9 Illustration of ROI retrieval on Kaggle Taxi data.

where P_i is a GPS point. By setting $\epsilon(l) = l \times 0.005 \times (\text{width} + \text{height})$, $l = \{1, \dots, 10\}$, we generate a 10-layer ROI network with CASCadesync. Fig. 8 shows the number and MSE of ROIs with different ϵ on each layer. The results manifest the information loss is linearly proportional to the ϵ . The most salient point is our hierarchical ROI network well preserves information from fine-grained to coarse-grained. Besides, the resemblance among the shape of curves manifests the insensitivity of parameter ϵ . For illustration, we visualize a 4-layer ROI network on Geolife in Fig. 5.

Table 3 NDCG@5 results of ROI retrieval. Model 1 only considers geometric neighborhoods. Model 2 considers geometric and basic semantic neighborhoods. Model 3 is trained with domain knowledge.

DataSets	Structure	Random	Model 1	Model 2	Model 3
Geolife	One-layer	0.429 ± 0.24	0.526 ± 0.18	0.560 ± 0.15	0.713 ± 0.18
	Hierarchical		0.581 ± 0.15	0.615 ± 0.13	0.724 ± 0.17
T-Drive	One-layer	0.382 ± 0.23	0.592 ± 0.21	0.521 ± 0.18	0.713 ± 0.21
	Hierarchical		0.650 ± 0.24	0.655 ± 0.23	0.693 ± 0.18
Kaggle	One-layer	0.410 ± 0.28	0.524 ± 0.20	0.542 ± 0.22	0.679 ± 0.19
	Hierarchical		0.524 ± 0.22	0.567 ± 0.21	0.691 ± 0.16
Chengdu	One-layer	0.415 ± 0.25	0.666 ± 0.20	0.677 ± 0.21	0.817 ± 0.17
	Hierarchical		0.697 ± 0.23	0.665 ± 0.20	0.802 ± 0.16

4.3.2 Semantic Retrieval with Different Embedding Schemes

Given a ROI, we retrieval the top-5 most similar ROIs, and NCDG@5 is computed. Besides, given a trajectory, we retrieval the top-10 most similar trajectories, and compare the performance with comparison metrics. Here we report

Table 4 Evaluation results of semantic trajectory retrieval on four dataset. The abbreviation **D.K.** denotes domain knowledge.

DataSet	Metric	Random	DTW	Frechet	LCSS	EDR
Geolife	NDCG@3	0.254 ± 0.158	0.391 ± 0.198	0.328±0.179	0.360 ± 0.258	0.340 ± 0.189
	NDCG@5	0.272 ± 0.130	0.410 ± 0.185	0.393±0.178	0.368 ± 0.224	0.411 ± 0.171
	NDCG@10	0.293 ± 0.127	0.521 ± 0.134	0.439±0.165	0.379 ± 0.184	0.467 ± 0.171
T-Drive	NDCG@3	0.286 ± 0.182	0.416 ± 0.201	0.304±0.191	0.396 ± 0.241	0.358 ± 0.195
	NDCG@5	0.290 ± 0.172	0.430 ± 0.168	0.346±0.156	0.402 ± 0.211	0.363 ± 0.184
	NDCG@10	0.343 ± 0.169	0.443 ± 0.149	0.371±0.141	0.402 ± 0.180	0.379 ± 0.161
Kaggle	NDCG@3	0.255 ± 0.220	0.311 ± 0.223	0.250±0.175	0.349 ± 0.227	0.290 ± 0.182
	NDCG@5	0.276 ± 0.225	0.321 ± 0.215	0.288±0.151	0.375 ± 0.218	0.304 ± 0.177
	NDCG@10	0.307 ± 0.201	0.342 ± 0.207	0.300±0.148	0.382 ± 0.214	0.308 ± 0.162
Chengdu	NDCG@3	0.303 ± 0.210	0.417 ± 0.168	0.360±0.239	0.314 ± 0.241	0.416 ± 0.196
	NDCG@5	0.320 ± 0.208	0.442 ± 0.153	0.388±0.229	0.346 ± 0.221	0.424 ± 0.171
	NDCG@10	0.345 ± 0.205	0.457 ± 0.145	0.417±0.211	0.376 ± 0.209	0.449 ± 0.179
DataSet	Metric	One-layer Model		Hierarchical Model		
		Without D.K.	With D.K.	Without D.K.	With D.K.	
Geolife	NDCG@3	0.468 ± 0.177	0.607 ± 0.152	0.545 ± 0.171	0.663 ± 0.148	
	NDCG@5	0.483 ± 0.165	0.639 ± 0.120	0.552 ± 0.162	0.676 ± 0.104	
	NDCG@10	0.520 ± 0.156	0.657 ± 0.109	0.558 ± 0.148	0.689 ± 0.094	
T-Drive	NDCG@3	0.529 ± 0.169	0.697 ± 0.179	0.547 ± 0.182	0.717 ± 0.184	
	NDCG@5	0.545 ± 0.144	0.711 ± 0.172	0.564 ± 0.169	0.702 ± 0.183	
	NDCG@10	0.544 ± 0.139	0.734 ± 0.168	0.599 ± 0.155	0.721 ± 0.175	
Kaggle	NDCG@3	0.406 ± 0.185	0.593 ± 0.164	0.438 ± 0.183	0.600 ± 0.209	
	NDCG@5	0.413 ± 0.160	0.604 ± 0.150	0.445 ± 0.151	0.602 ± 0.183	
	NDCG@10	0.430 ± 0.150	0.607 ± 0.157	0.452 ± 0.145	0.626 ± 0.188	
Chengdu	NDCG@3	0.588 ± 0.159	0.824 ± 0.143	0.611 ± 0.160	0.833 ± 0.120	
	NDCG@5	0.622 ± 0.124	0.859 ± 0.139	0.635 ± 0.139	0.849 ± 0.112	
	NDCG@10	0.651 ± 0.112	0.866 ± 0.113	0.666 ± 0.114	0.862 ± 0.096	

NCDG@3,5,10, respectively. Moreover, we set the number of each neighborhood as 3, and the negative sampling number as 5, and $\epsilon(l) = l \times 1/200 \times (width + height)$, $l = \{1, \dots, 5\}$ to derive a 5-layer hierarchical ROI network.

To explore the effects of different semantic information on real-world data, we vary the types of neighborhoods incorporated into embedding model in three ways: 1) Only geometric information (i.e., basic trajectory structure); 2) Geometric and basic semantics (i.e., network features and temporal information); 3) All kinds of information and domain knowledge are modeled. Moreover, we investigate the effect of feature propagation by constructing two embedding models: Embedding model trained only on the bottom layer ($\epsilon(t) = 1/200 \times (width + height)$) of hierarchical ROI network via (8) and the whole model on all five layers via (10).

For better illustration, we select a ROI represented a hub junction in Kaggle Taxi data as a target ROI. We apply the embedding models with/without semantic neighborhoods on the aforementioned 5-layer network, the retrieval results are shown in Fig. 9. From the results, the model with semantic neighborhoods allows capturing the semantic information and thus easily identifies

the urban roles of regions, and retrieves hub junctions far away from querying target junction. The reason is that the semantics of traffic is reflected by degree and distribution of neighbors' degree of ROIs on the network, thereby, is not limited by its geometric location.

The results of ROIs retrieval and trajectories retrieval are summarized in Table 3 and Table 4 respectively. From the tables, we find that: (1) basically, the more semantic neighborhoods are considered in the embedding models, the better the performance is; (2) when there is no domain knowledge, embedding models on hierarchical ROI network are superior to one-layer version, which means the hierarchical structure boosted the embedding by feature propagation. (3) when embedding model trained with domain knowledge, the performance relies mainly on label quality.

5 Conclusion

In this paper, we propose a new semantic trajectory representation. To start with, the raw trajectories are summarized as a multi-resolution ROI network via synchronization-based clustering model: CASCadesync. Relying on the network, by extracting the geometric, semantic, temporal information as well as additional domain knowledge as context information, ROIs and trajectories are learned as continuous vectors in a semantic space via distributed vector representation, whose metric is tailored for measuring the semantic similarity. Building upon the derived vectors, semantic retrieval tasks can be effectively and efficiently performed by computing the Euclidean distance of embedding vectors directly. Empirical experiments on both synthetic and real-world data sets indicate that our proposed approach allows extracting semantic information effectively, and outperforms classical metrics: DTW, LCSS, EDR and Fréchet distance, on the semantic trajectory retrieval tasks.

References

- Agrawal R, Faloutsos C, Swami A (1993) Efficient similarity search in sequence databases. In: Foundations of Data Organization and Algorithms
- Alvares LO, Bogorny V, Kuijpers B, Moelans B, Fern JA, Macedo E, Palma AT (2007) Towards semantic trajectory knowledge discovery. In: DMKD'07
- Anagnostopoulos A, Atassi R, Beccetti L, Fazzone A, Silvestri F (2017) Tour recommendation for groups. *Data Mining and Knowledge Discovery* 31(5):1157–1188
- Annoni R, Forster CH (2012) Analysis of aircraft trajectories using fourier descriptors and kernel density estimation. In: ITSC'12
- Ardakani IS, Hashimoto K (2017) Encoding bird's trajectory using recurrent neural networks. In: ICMA'17
- Bellman R (1961) On the approximation of curves by line segments using dynamic programming. In: Communications of the ACM

- Bengio Y, Ducharme R, Vincent P, Jauvin C (2003) A neural probabilistic language model. In: JMLR'03
- Blacoe W, Lapata M (2012) A comparison of vector-based representations for semantic composition. In: EMNLP'12
- Bogorny V, Kuijpers B, Alvares LO (2009) St-dmql: A semantic trajectory data mining query language. In: International Journal of Geographical Information Science
- Böhm C, Plant C, Shao J, Yang Q (2010) Clustering by synchronization. In: SIGKDD'10
- Chakka VP, Everspaugh AC, Patel JM (2003) Indexing large trajectory data sets with seti. In: CIDR'03
- Chen L, Özsü MT, Oria V (2005) Robust and fast similarity search for moving object trajectories. In: SIGMOD'05
- Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa P (2011) Natural language processing (almost) from scratch. In: JMLR'11
- Eiter T, Mannila H (1994) Computing discrete fréchet distance. Tech. rep.
- Gao Q, Zhou F, Zhang K, Trajcevski G, Luo X, Zhang F (2017) Identifying human mobility via trajectory embeddings. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence, AAAI Press, pp 1689–1695
- Gariel M, Srivastava AN, Feron E (2011) Trajectory clustering and an application to airspace monitoring. In: Transactions on Intelligent Transportation Systems
- Iyyer M, Manjunatha V, Boyd-Graber J, Daumé III H (2015) Deep unordered composition rivals syntactic methods for text classification. In: ACL-IJCNLP'15
- Kalchbrenner N, Grefenstette E, Blunsom P (2014) A convolutional neural network for modelling sentences. In: arXiv preprint arXiv:1404.2188
- Kavraki LE, Svestka P, Latombe JC, Overmars MH (1996) Probabilistic roadmaps for path planning in high-dimensional configuration spaces. In: IEEE Transactions on Robotics and Automation
- Kellaris G, Pelekis N, Theodoridis Y (2009) Trajectory compression under network constraints. In: SSTD'09
- Kim CS, Bae CS, Tcha HJ (2008) A phase synchronization clustering algorithm for identifying interesting groups of genes from cell cycle expression data
- Lee JG, Han J, Whang KY (2007) Trajectory clustering: A partition-and-group framework. In: SIGMOD'07
- Lee WC, Krumm J (2011) Trajectory preprocessing. In: Computing with Spatial Trajectories
- Li Q, Zheng Y, Xie X, Chen Y, Liu W, Ma WY (2008) Mining user similarity based on location history. In: SIGSPATIAL'08
- Li Z, Ding B, Han J, Kays R, Nye P (2010) Mining periodic behaviors for moving objects. In: SIGKDD'10
- Mazimpaka JD, Timpf S (2016) Trajectory data mining: A review of methods and applications. In: Journal of Spatial Information Science

- Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: NIPS'13
- Mitchell J, Lapata M (2010) Composition in distributional models of semantics. In: Cognitive Science
- Nascimento MA, Silva JR (1998) Towards historical r-trees. In: SAC'98
- Pelekis N, Tampakis P, Vodas M, Doulkeridis C, Theodoridis Y (2017) On temporal-constrained sub-trajectory cluster analysis. Data Mining and Knowledge Discovery 31(5):1294–1330
- Pennington J, Socher R, Manning C (2014) Glove: Global vectors for word representation. In: EMNLP'14
- Pfoser D, Jensen CS, Theodoridis Y, et al (2000) Novel approaches to the indexing of moving object trajectories. In: VLDB'00
- Seliger P, Young SC, Tsimring LS (2002) Plasticity and learning in a network of coupled phase oscillators. In: Physical Review E
- Shao J, He X, Böhm C, Yang Q, Plant C (2013) Synchronization-inspired partitioning and hierarchical clustering
- Shao J, Han Z, Yang Q, Zhou T (2015) Community detection based on distance dynamics. In: SIGKDD'15
- Shao J, Gao C, Zeng W, Song J, Yang Q (2017) Synchronization-inspired co-clustering and its application to gene expression data. In: ICDM'17
- Shokoohi-Yekta M, Hu B, Jin H, Wang J, Keogh E (2017) Generalizing dtw to the multi-dimensional case requires an adaptive approach. Data mining and knowledge discovery 31(1):1–31
- Socher R, Huang EH, Pennin J, Manning CD, Ng AY (2011) Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In: NIPS'11
- Song R, Sun W, Zheng B, Zheng Y (2014) Press: A novel framework of trajectory compression in road networks. In: VLDB'14
- Tai KS, Socher R, Manning CD (2015) Improved semantic representations from tree-structured long short-term memory networks. In: ACL'15
- Vlachos M, Kollios G, Gunopulos D (2002) Discovering similar multidimensional trajectories. In: ICDE'02
- Wieting J, Bansal M, Gimpel K, Livescu K (2016) Towards universal paraphrastic sentence embeddings. In: ICLR'16
- Yi BK, Jagadish H, Faloutsos C (1998) Efficient retrieval of similar time sequences under time warping. In: ICDE'98
- Yuan J, Zheng Y, Zhang C, Xie W, Xie X, Sun G, Huang Y (2010) T-drive: driving directions based on taxi trajectories. In: Proceedings of the 18th SIGSPATIAL International conference on advances in geographic information systems
- Yuan J, Zheng Y, Xie X, Sun G (2011) Driving with knowledge from the physical world. In: Proceedings of the 17th ACM SIGKDD international conference on knowledge discovery and data mining, ACM, pp 316–324
- Yuan NJ, Zheng Y, Zhang L, Xie X (2013) T-finder: A recommender system for finding passengers and vacant taxis. In: TKDE'13
- Zheng Y, Xie X (2011) Learning travel recommendations from user-generated gps traces. In: TIST'11

- Zheng Y, Li Q, Chen Y, Xie X, Ma WY (2008a) Understanding mobility based on gps data. In: Proceedings of the 10th International Conference on Ubiquitous Computing
- Zheng Y, Liu L, Wang L, Xie X (2008b) Learning transportation mode from raw gps data for geographic applications on the web. In: WWW'08
- Zheng Y, Zhang L, Xie X, Ma WY (2009) Mining interesting locations and travel sequences from gps trajectories. In: WWW'09
- Zheng Y, Xie X, Ma WY (2010) Geolife: A collaborative social networking service among user, location and trajectory. *IEEE Data Eng Bull* 33(2):32–39