Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Find you if you drive: Inferring home locations for vehicles with surveillance camera data

Kai Chen[b], Yanwei Yu[a,b,*], Peng Song[b], Xianfeng Tang[c], Lei Cao[d], Xiangrong Tong[b]

[a] Department of Computer Science and Technology, Ocean University of China, 238 Songling RD, Qingdao, Shandong 266100, China
[b] School of Computer and Control Engineering, Yantai University, 30 Qingquan RD, Yantai, Shandong 264005, China
[c] College of Information Sciences and Technology, Pennsylvania State University, 201 Old Main, University Park, PA 16802, USA
[d] CSAI Lab, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139, USA

ARTICLE INFO

Article history:
Received 17 June 2019
Received in revised form 30 January 2020
Accepted 9 March 2020
Available online 12 March 2020

Keywords:
Urban computing
Surveillance camera data
Home location inference
Kernel density estimation
Vehicle trajectory

ABSTRACT

Inferring home locations for users from spatiotemporal data has become increasingly important for real-world applications ranging from security, recommendation, advertisement targeting, to transportation scheduling. Existing home location inference studies are based either on geo-tagged social media data or continuous GPS data. Yet this inference problem in highly sparse vehicle trajectories in urban surveillance systems remains largely unexplored. In this paper, we propose an accurate home location inference framework for vehicles in urban traffic surveillance systems by considering both spatial and temporal characteristics. To the best of our knowledge, we are the first to predict exact home community for vehicles at such a fine granularity using the sparse and noisy surveillance camera data. First, we collect and preprocess multiple contextual datasets to obtain a context-rich road network with residential communities and surveillance cameras. Second, we detect the potential home location areas for each vehicle by clustering Origin–Destination (*O-D*) pairs extracted in vehicle's camera-based trajectories. Then we further propose an *in*/*out* time pattern to distinguish the home area candidate from the *O-D* clusters by leveraging time-aware constraints. Furthermore, to find the exact home community, we propose a Kernel Density Estimation (KDE) based inference method with a local camera selection strategy to effectively identify the home community from the residential communities near/in the home area candidate. Our comprehensive experiments on a large-scale real-world dataset demonstrate the effectiveness of our proposed method.

© 2020 Elsevier B.V. All rights reserved.

1. Introduction

Motivation. Nowadays, spatiotemporal trajectory data can be collected from a variety of sources including location-sharing social networks (e.g., Foursquare check-ins), geo-tagged social media (e.g., Twitter and Weibo), location-based online services (e.g., Uber and Didi), and urban traffic surveillance systems (e.g., surveillance cameras and vehicle-mounted GPS). These spatiotemporal data provide us with a new dimension to understand human behaviors in the physical space and further benefit every aspect of life, such as transportation, healthcare, urban planning, and homeland security. Among these, inferring home locations for users has become increasingly important for real-world applications, such as security, localized recommendation, advertisement targeting, and transportation scheduling. For example, if we infer that two users live in the same residential community, it may imply that they have similar life demands, then we can recommend them to become online friends or suggest them carpool when one user goes to the places that the other user frequently visits. In addition, if we know the home location of a vehicle, we can easily understand each mobility for the vehicle and thus sense the individual behavior patterns. Many studies have been conducted to infer home location based on users' spatiotemporal data in areas of location-based social networks [1–7], online social media [8–14], and dense GPS trajectories [15–18].

Although extensive research has been done to infer home location from spatiotemporal data, the existing methods still have two key limitations for inferring home location for vehicles. First, the links of online users and drivers are unknown, thus the methods based on social networks and social media cannot be used to infer home location for vehicles. Second, although the methods based on GPS trajectories have high prediction accuracy, for government departments and managers, the GPS trajectories of private vehicles are not easily available due to privacy issues.

* Correspondence to: 238 Songling RD, Laoshan District, Qingdao, Shandong 266100, China.
E-mail addresses: ytuchenkai@163.com (K. Chen), yuyanwei@ouc.edu.cn (Y. Yu), pengsongseu@gmail.com (P. Song), xianfeng@ist.psu.edu (X. Tang), lcao@csail.mit.edu (L. Cao), txr@ytu.edu.cn (X. Tong).

https://doi.org/10.1016/j.knosys.2020.105766
0950-7051/© 2020 Elsevier B.V. All rights reserved.

In recent years, surveillance cameras have been widely deployed to monitor traffic situations. These AI-equipped cameras can recognize individual vehicle information (e.g., license plate, speed, driving direction, etc.). Therefore, vehicles' License Plate Recognition (LPR) data are available for almost all kinds of vehicles no matter they have GPS devices installed or not in urban surveillance systems. The properties of surveillance cameras enable us to consider home location inference for all vehicles based on the surveillance camera data. Although the drivers' personal information is filed in the department of transportation, most of drivers do not live at the registered addresses due to multiple reasons such as changing rental address, collective registered residence, multiple housing, and off-site work. Home location inference for vehicles with surveillance cameras benefits varieties of applications ranging from homeland security, traffic scheduling, to urban planning. Although surveillance camera data is not available for users and service providers, government departments can also provide service providers with query interfaces to support more location-based recommendation and advertising services.

In this paper, we are interested in inferring the home location for vehicles with surveillance camera data in urban traffic surveillance systems. Despite the importance of inferring home locations for vehicles in urban management, to the best of our knowledge, this problem has not been considered in the previous literature.

Challenges. Despite the growing adoption of traffic surveillance cameras, their coverage in the city is still limited because of the cost of installment and maintenance. The observed trajectories are incomplete because they are obtained from the static and discrete cameras. Therefore, the problem of home location inference with surveillance camera data faces multiple challenges:

- **Sparsity.** Surveillance cameras only cover partial intersections and road segments in urban. A private vehicle only is recorded at a few cameras on some days in a city. Therefore, vehicle trajectories based on surveillance camera data are incomplete and highly sparse in both spatial and temporal dimensions.
- **Noisy.** The collected data is extremely "noisy". For example, a vehicle may have different starting areas on different days, or inconsistent starting area and ending area on same day. In addition, we find that some surveillance cameras may not work on some days, which causes inconsistent camera records.
- **Static.** The locations of surveillance camera are fixed, thus the vehicles only are observed at fixed locations (usually at intersections) on road network. That is, the stay points in the vehicle trajectories are static with respect to the locations of cameras.
- **Too many nearby residential communities.** Since the locations of cameras are mainly located at road intersections, there are many residential communities nearby. Moreover, a vehicle may be observed at multiple starting and ending areas, which makes it more challenging to infer the true home location. In fact, the nearest residential community may not be the home location of vehicle due to the sparsity of vehicle trajectories and partial coverage of surveillance cameras.

In literature, a variety of studies has been done to predict home location based users' check-in data and/or textual contents in social media [8,19–26]. Basically, these methods apply supervised learning to predict users' home location based on the features of check-ins, places extracted from textual contents, and user profile. But most methods only achieve coarse-grained location inference in levels of town, city, post-code or state with a large error. Another line of studies utilizes the locations of users' friends to infer their home locations in Location-Based Social Networks (LBSNs) [1,2,4,5,7]. These work leverages the social relationships and partial locatable friends to infer users' home location. However, such methods require the social relationships between users and partial ground truth, which both cannot be directly apply to surveillance camera data.

Recently, several studies focus on fine-grained semantic location inference based GPS trajectory [15,16,18,27–29]. In [15], four heuristic algorithms (i.e., Last Destination, Weighted Median, Largest Cluster and Best Time) are proposed to infer uses' home location in GPS trajectories, with a median error of 60 m. Wan et al. [17] propose to mine spatial–temporal semantic mobility patterns from trajectories of private vehicles based on their GPS data and POI data. These two approaches cannot be applied to our problem because they fail to handle the sparsity and static challenges in vehicles' LPR data. To annotate mobility data, Wu et al. [28] propose to capture the relevant semantic words with respect to a mobility record using contextual social media. In their follow-up work, Wu et al. [18,29] attempt to understand taxi traffic dynamics from multiple external data sources including POI, weather, geo-tagged tweet, and collision records. They propose to use ridge regression with polynomial kernel to describe the non-linear non-additive relationships of impacting factors. However, in our problem, it is impossible to match and extract the meaningful textual information with LPR data from noisy external social media due to the sparsity and stationary of deployed surveillance cameras. Therefore, these approaches also cannot be applied to our problem.

Proposed solution. To address the aforementioned challenges, this article proposes a novel home location inference framework for vehicles in surveillance camera data by considering both spatial and temporal characteristics. First, we obtain a real-world road network with residential communities and surveillance cameras by projecting collected multiple contextual data to road network. Second, we propose a new discovery method to detect the potential home location areas for each vehicle by clustering Origin–Destination (*O-D*) pairs in its camera-based trajectories. Specifically, we propose an *in*/*out* time pattern to distinguish the home area candidate from the *O-D* clusters by leveraging time-aware constraints. Third, to find the home community, we further propose a KDE-based inference method to effectively detect the home community from the residential communities near/in the home area candidate. To improve prediction accuracy, we finally propose a local camera selection strategy to choose the suitable local cameras for each community candidate in KDE-based model.

We use a large-scale real-world dataset collected from a provincial capital in China for a whole month of August in 2016. There are more than 11 million unique vehicles with about 405 million camera records. We conduct comprehensive experiments to demonstrate the effectiveness of our proposed method.

To summarize, we make the following contributions:

- We are the first to propose and formally define the problem of home location inference for vehicles with surveillance camera data in urban traffic surveillance systems.
- We propose a novel home area candidate discovery method to detect the largest possible home areas for vehicles by clustering *O-D* pairs extracted from vehicle trajectories and matching them with time-aware constraints.
- We propose an effective home community inference method using KDE to model the density of residential community with respect to vehicle passing local cameras. We design a local camera selection strategy to better choose suitable local cameras for each community candidate in KDE-based model.

- We conduct extensive evaluations on a large-scale real-world dataset. Experimental results demonstrate the effectiveness of our proposed method.

2. Related work

2.1. Home location inference in social media

There have been many studies on how to infer the home locations for users in social media. A line of methods have been proposed to predict users' city-level location in Twitter based purely on the content of users' tweets [8,19–26]. Mahmud et al. [24,30] present a method that uses an ensemble of statistical and heuristic classifiers to predict locations and makes use of a geographic gazetteer dictionary to identify place-name entities for inferring the home location of Twitter users at different levels of granularity, including city, state, time zone or geographic region, using the content of users' tweets and their tweeting behavior. Schulz et al. [8] use the names of places that appear in the text message, dedicated location entries and additional information from the user profile to determine the location where a tweet was created and the location of the user's residence. Li et al. [9] propose a location identification method for inferring top-k city-level locations of a user from her microblogs. Kondo et al. [25] propose a method to estimate a Twitter user's home location area by incorporating observed weather data and user's tweet contents. Miura et al. [26,31] use the deep learning techniques to predict home city for users based on tweet contents. Such methods only achieve coarse-grained location inference for users in levels of town, city, and/or state, even country.

There is a variety of studies that tries to utilize users' check-in data to infer their home locations [11–14]. Efstathiades et al. [10] propose a simple method for identifying a user's key locations (i.e., her home and work places) from geo-tagged tweets. Users may spend more time at home or at workplace. The location of home or workplace is inferred according to the location of tweets in different time periods. However, the method only achieves about 60% accuracy at post-code level. Hu et al. [11] use a linear SVM model to infer the users' home locations using multiple features of their check-in data, such as check-in rate, check-in rate during midnight, last destination with inactive midnight, and so on. That is, they transfer the inference problem into an equivalent classification problem. In a follow-up work, Kavak et al. [32] further extend [11] by adding two mobility features (i.e., land use patterns and distances from most checked-in location). Similarly with [11,32], Hossain et al. [12] also train a SVM classifier to predict home location for active users within 100 by 100 m grids from geo-tagged tweets. Poulston et al. [13] propose a cluster-based approach to predict user's hyperlocal home location from her geo-tagged tweets. The most populous cluster is taken as the users' home, and its geometric median is taken as the user's home coordinate. Lin et al. [14] use four simple method (i.e., the weighted most frequently visited, the weighted-mean, weighted-median and the support vector machine) based on temporal and spatial features from geo-tagged tweets to identify a Twitter user's home location.

A variety of researches that utilizes the locations of users' friends to infer their home locations in Location-Based Social Networks (LBSNs) have been conducted [1,2,4,5,7]. [1,2,6,33,34] aim to predict users' city-level home location using social relationships and their locatable friends. Backstrom et al. [35] propose an algorithm to predict the physical location of a user, given the known location of her friends. More specifically, they measure the relationship between proximity and friendship, and predict the locations of individuals based on the propagation of predictions across the network. [4,36,37] also use label propagation approach to infer users' locations through social networks. Hicham G. et al. [38] propose a graph-based location inference model to infer the users' home location based on both their social graph and tweets content. Li et al. [3] propose a unified discriminative influence model (UDI) that integrates signals observed from both user-centric data and friendships to profile users' home locations in context of social networks. In the follow-up work, [5,39] further propose trust-based influence model in social networks to improve inference performance. McGee et al. [40] propose a network-based approach for location estimation in social media that integrates evidence of the social tie strength and distance between users for improving location estimation. Kong et al. [41] propose three location estimation methods based on social network context for Twitter users. Huang et al. [42] propose an unsupervised home location inference that explores the spatial, temporal and social relationship dimensions of users to infer home locations of people in LBSNs. In their follow-up work [7], they also propose a semi-supervised framework to infer the home locations of users by explicitly exploring the localness of users and the dependency between users based on their check-in behaviors.

However, these home location inference methods based on check-in data in social media cannot be applied into our problem, since they only focus on determining one location as user's home from given check-in locations.

2.2. Location inference from GPS trajectories

Several studies focus on refined location inference based dense GPS trajectory data [15–17,43,44]. Krumm et al. [15] use four heuristic algorithms (i.e., Last Destination, Weighted Median, Largest Cluster and Best Time) to find users' home location based on the GPS trajectories, with a median error of 60 m. Zheng et al. [27] propose a tree-based hierarchical graph with multiple individuals' location histories to mine the interesting locations and travel sequences based on users' GPS trajectories. Xiao et al. [44] propose a method that models a user's GPS trajectories with a semantic location history via stay point detection, e.g., shopping malls $\longrightarrow$ restaurants $\longrightarrow$ cinemas. Cao et al. [16] propose a method to extract significant semantic locations from GPS data. More specifically, they capture the relationships between locations and between locations and users with a graph. Significance is then assigned to locations using random walks over the graph that propagates significance among the locations. Wan et al. [17] propose an approach to mine spatial–temporal semantic mobility patterns from trajectories of private vehicles based on the vehicles' GPS data and POI data. They design a probabilistic generative model with latent variables to characterize the semantic mobility of vehicles. But these methods are not suitable for inferring home location of vehicles in traffic surveillance system because GPS trajectories are highly more dense than surveillance cameras.

2.3. Semantic exploration for mobility data

Zheng et al. [45] propose a probabilistic framework which uncovers and quantifies characteristic behavior patterns in user's daily lives from mass amount of mobile data in unsupervised setting and exploits it to predict user activities. Yuan et al. [46] propose a probabilistic model W4 (short for Who + Where + When + What) to exploit geo-annotated tweet data to discover individual user's mobility behaviors from spatial, temporal and activity aspects. The model has a variety of applications, such as user profiling and location prediction. Krumm et al. [47] propose a multiclass classifier in the form of a forest of boosted

Table 1
Notations and descriptions.

Notation	Description
veh_{id}	A vehicle with license plate number *id*.
cam_i	A camera.
ts_i	A timestamp.
Tr_{id}	A vehicle trajectory of veh_{id}.
TR	The set of all vehicle trajectories of veh_{id}.
$\mathcal{G} = (N, E)$	A road network.
N	The set of intersections in $\mathcal{G}$.
E	The set of road segments in $\mathcal{G}$.
(cmt_{id}, G)	A residential community cmt_{id} with its gate set G.
g_i	An entrance and exit of a residential community.
CMT	The set of all residential communities.
Ω	The set of local origin cameras in the home area.
Φ	The set of local destination cameras.
h	The bandwidth.
c_i	A local camera.
w_{c_i}	The weight of the *i*th local camera.
τ	The time interval threshold.
θ	The visit frequency threshold.
γ	The selected block threshold.

decision trees to infer semantic places labels based on individual demographic features and temporal features of the visits. Wu et al. [28] propose frequency-based method, Gaussian mixture model and kernel density estimation to capture the relevant semantic words with respect to a mobility record to annotate mobility data using contextual social media. In their follow-up work, Wu et al. [18,29] attempt to understand taxi traffic dynamics from multiple external data sources including POI, weather, geo-tagged tweet, and collision records. They propose to use ridge regression with polynomial kernel to describe the non-linear non-additive relationships of impacting factors. Hu et al. [48] intend to decode human life from two perspectives: linguistic perspective and mobility perspective to mine knowledge of human life, including urban and regional lifestyles and shopping patterns. From a mobility perspective, they extract the mobility patterns of individual person, and groups of people such as residents of certain regions. However, these semantic modeling methods require accurate location information and annotated social media, which does not apply to sparse surveillance camera data without contextual text information.

3. Problem definition

In this section, we define the key notations used in the paper and then formally define our problem. We define camera record as follows:

Definition 1 (*Camera Record*)**.** A camera record is a triple (veh_{id}, cam_i, ts_j) that represents vehicle veh_{id} passing through camera cam_i at timestamp ts_j.

Definition 2 (*Vehicle Trajectory*)**.** The trajectory of a vehicle veh_{id} is a sequence of tuples in chronological order, denoted by $Tr_{id} = \{\langle cam_1, ts_1\rangle, \langle cam_2, ts_2\rangle, \ldots, \langle cam_i, ts_i\rangle, \ldots, \langle cam_n, ts_n\rangle\}$, where each tuple $\langle cam_i, ts_i\rangle$ indicates veh_{id} passing through camera cam_i at timestamp ts_i.

Note that we treat the camera records of each vehicle on one day as a vehicle trajectory, and all vehicle trajectories of a vehicle form the vehicle trajectory set *TR*. We use *TRs* to denote the set of all vehicle trajectories of all vehicles.

From Definitions 1 and 2, we learn that the vehicle trajectories of vehicles are based on the installed surveillance cameras in a city, thus such trajectories are incomplete and highly sparse.

Definition 3 (*Road Network*)**.** A road network is denoted as $\mathcal{G} = (N, E)$, where $N = \{n_1, n_2, \ldots, n_m\}$ is the set of intersections, E is the set of road segments between the intersections, and each edge $e_{i,j} \in E$ represents the road segment from intersection n_i to intersection n_j.

Note that each road segment is directed, i.e., road segment $\langle n_i, n_j\rangle$ is different from road segment $\langle n_j, n_i\rangle$ because of the different directions.

Moreover, each camera is fixed at one exact road segment, namely, the vehicles captured by camera cam_i pass through its corresponding road segment. Therefore, each camera record also implies the passing direction of the vehicle.

Definition 4 (*Residential Community*)**.** A residential community is denoted as (cmt_{id}, G), where cmt_{id} is the identifier of the residential community and G is the set of community gates, each gate g_i representing an entrance and exit of the residential community with the geographical coordinates (i.e., longitude and latitude).

All collected residential communities form the set of residential communities, denoted as CMT.

Finally, we formally define our problem as follows:

Problem 1 (*Home Location Inference for Vehicles*)**.** Given a road network $\mathcal{G}$, vehicles' trajectory set *TRs* and residential community set CMT, our goal is to infer the residential community with highest probability as the home location for each vehicle.

The major notations used throughout the paper are summarized in Table 1.

4. Home location inference framework

4.1. Overview

Fig. 1 shows the overall framework of our proposed home location inference method. Our framework is mainly composed of three parts: data preprocessing, home area candidate discovery and KDE-based home community inference. The data preprocessing mainly includes extracting vehicle trajectories from surveillance camera data, and matching cameras and residential communities with road network. The second part aims to find the candidate area of home location for each vehicle by clustering stay points (e.g., starting cameras and ending cameras) in the camera-based vehicle trajectories. Home community inference module is to calculate the probability that the vehicle belongs to each residential community in the discovered home area candidate.

4.2. Data preprocessing module

4.2.1. Vehicle trajectory extraction

In surveillance camera data, each record represents a vehicle passing through a camera at a timestamp. According to the definition of vehicle trajectory, to get the trajectories for each vehicle, we can track the vehicle via its unique license plate, and rank the camera records created by the vehicle in chronological order. Common sense tells us that people's mobility activities often have a daily cyclical character (i.e., periodicity), which is proved in many studies [49,50]. Therefore, we treat the trajectory of a vehicle on each day as a vehicle trajectory, thus all vehicle trajectories of a vehicle forms its vehicle trajectory set.

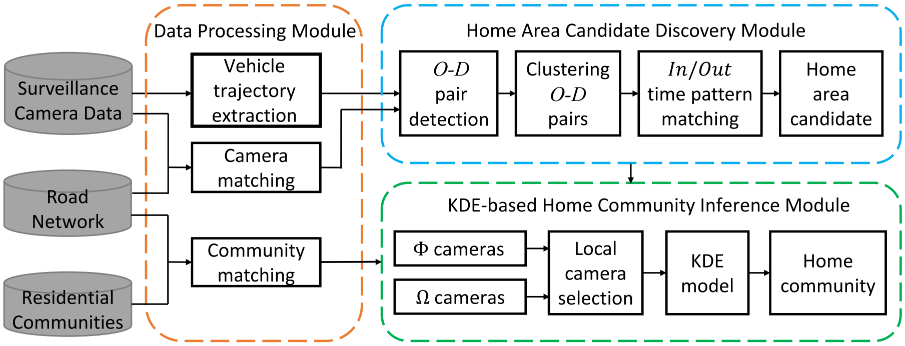


Fig. 1. Overall framework.

4.2.2. Camera matching

We obtain the road network from the *OpenStreatMap* [51], which is public available. As defined in Definition 3, a road network includes a set of intersections and a set of road segments between these intersections. Generally, surveillance cameras are deployed near intersections and are capable of monitoring all passing vehicles from one direction. That is, the vehicles captured by one camera pass through its fixed road segment. Therefore, we match the cameras with collected road network according to the geographical locations of cameras (e.g., latitude and longitude) to obtain the corresponding road segment for each camera. In this way, we can obtain the incomplete vehicle trajectories and driving directions of all vehicles on road network.

4.2.3. Residential community matching

We crawl the residential communities from Baidu Map. More specifically, a residential community may have multiple gates for the entrance and exit of vehicles. We use the community gate information to represent each residential community as defined in Definition 4. To support the road network distance calculation, we match the community gates with road network using the latitude, longitude and direction of the community gates. Each gate is mapped to an exact nearest road segment. Therefore, a residential community is located by its all gates in the road network. Finally, we obtain a context-rich road network with residential communities and surveillance cameras.

4.3. Home area candidate discovery

In this section, we introduce how to select the candidate area of home location for each vehicle. This part mainly contains three steps: (1) detecting Origin–Destination (*O-D*) pairs in vehicle trajectories, (2) clustering the detected *O-D* pairs, and (3) discovering the home area candidate with time-aware constraints.

4.3.1. O-D pair detection

People's most activities are built around the home location. For example, if a person wants to go shopping, he would start from home location and arrive at a mall, and then return home after shopping. Therefore, we focus more on the starting points and ending points in vehicle trajectories to discover the home location. Although the camera based trajectories are sparse with respect to time and space, we can still detect each independent sub-trajectories from the vehicle trajectories by detecting the stay points, which maximizes the use of the vehicle trajectories. In particular, we partition the vehicle trajectory into sub-trajectories by examining whether the time interval between two adjacent camera records exceeds a given time threshold τ. For each sub-trajectory, the starting point and ending point is regarded as an *O-D* pair. People always go out from home and then return home afterwards. Hence home location is usually implied near these starting and ending points. Therefore, we extract all *O-D* pairs from vehicle trajectory set *TR*. Note that the origin and destination in *O-D* pairs are cameras' locations in the vehicle trajectories.

4.3.2. Clustering O-D pairs

As we mentioned earlier, there is a lot of noise in the camera data. Vehicles may not be captured by the passing camera nearest to their home communities sometimes, which causes that there are multiple different origins and destinations in the same areas. To get the important areas that each vehicle frequently visits, we use DBSCAN clustering algorithm to group the neighboring cameras for each vehicle's *O-D* pairs. We use the location information of the cameras in the *O-D* pairs as input. Each obtained cluster that includes multiple cameras represents an area where the vehicle often visits, which may locate the exact home location of the vehicle. Note that we omit the origins and destinations with low frequency that is less than θ. Because vehicles may occasionally arrive at these places if the visit frequency is very low. In addition, we treat each visiting record as one input point, that is, for one origin or destination with visit frequency *freq*, we regard there are *freq* input points. We set $MinPts = 5$, $Eps = 1000$ m, meaning that the distance between any two cameras in each cluster is not larger than 1000 m and there are at least five visits to the cameras in the cluster.

4.3.3. Discovering home area with time-aware constraints

The clustering method only allows us to discover the important areas where the vehicle often visits, such as home location, workspace, shopping malls, etc. To further distinguish the home area from other areas, we further leverage time-aware constraints to refine the home location area candidate.

Usually, people always go work from home in the morning and return home at nightfall. Sometimes people may also go back home at noon for lunch or taking a rest, and continue to work in the afternoon. Therefore, we propose an *in/out* time pattern to present the visit time information for each cluster as destination and origin, respectively. Namely, *out* time pattern records the visit time of all cameras in each cluster as origin, and *in* time pattern corresponds to destination. If most time in *out* time pattern of a cluster is in the early morning and most time in corresponding *in* time pattern is at nightfall, the cluster is more likely to be the home area than other clusters with unstable patterns.

To quantify the time pattern, we use a Gaussian kernel-based KDE to estimate the distribution of visit time for each pattern. We first select the cluster that has one highest peak in the early morning or early afternoon in *out* time pattern and has one highest peak in the early evening or noon in *in* time pattern. If there are multiple clusters that satisfy the time constraints, we choose the cluster with the most frequency (i.e., the sum of frequency of all cameras) as home area candidate.

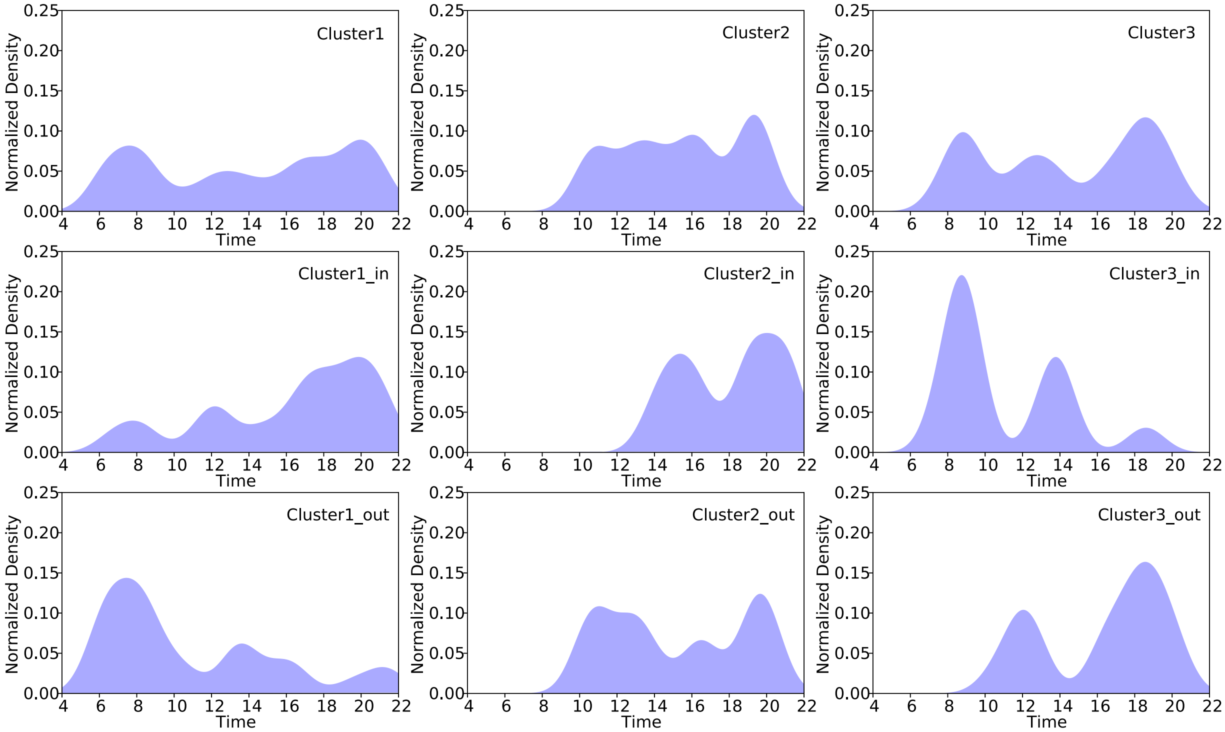


Fig. 2. An example of *in/out* time patterns.

Fig. 2 shows an example of *in/out* time patterns of three obtained biggest clusters for a given vehicle. First row shows the overall time distribution for three clusters, and the last two rows depict *in* and *out* time patterns respectively. We can see that *cluster1* (actual home area) has the highest peak around 7 AM in *out* time pattern and the highest peak between 6 PM and 8 PM in *in* time pattern. Although *cluster3* has similar overall time distribution with *cluster1*, it shows a completely different time distribution in *out* and *in* time patterns, respectively. Specifically, *cluster3* clearly shows the highest peak around 9 AM in *in* time pattern and the highest peak at about 6 PM in *out* time pattern. This is obviously in line with time distribution of workplace. Therefore, *cluster1* would be chosen as the home area candidate using *in/out* time pattern matching for the vehicle.

4.4. KDE-based home community inference

The home area candidate actually only contains the neighboring cameras near the home community. There may be multiple residential communities in the candidate area. Hence we first extract the residential communities near the cameras in the home area candidate as home community candidates.

Usually, residents frequently visit several cameras near their home communities when they go out to work or return home. Therefore, residents' visit records to the nearby cameras should satisfy a distribution centered on the residential community with respect to *distance* and *frequency*. With these considerations in mind, we propose to use Kernel Density Estimation (KDE) method to model the spatial density of the home community candidates with respect to their local cameras.

4.4.1. KDE model

KDE is a non-parametric model for estimating density from sample points. Following the kernel density model, we define the density score for a community x with respect to its local cameras as follows:

$$f_h(x) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - c_i) = \frac{1}{nh}\sum_{i=1}^{n} K(\frac{x - c_i}{h}), \tag{1}$$

where h represents the bandwidth, x stands for a community, c_i represents the sample i and n represents the number of samples. In our problem, the samples are the local cameras in the home area candidate, $(x - c_i)$ denotes the road network distance between the community x and the camera c_i, and $K(.)$ is the kernel function.

Generally, the closer a camera is to the community, the higher probability it is for residents to pass the camera. Therefore, we use the Gaussian kernel function in the paper, which is expressed as follows:

$$K(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}. \tag{2}$$

From Eq. (1), we can find that the closer the local cameras are to a community, the higher the density value of the community, that is, the higher the probability that the vehicle belongs to the community.

Because a community may have multiple gates, it is difficult to appropriately measure the distance between local camera and the community. We use the shortest distance from the camera to all gates to represent the distance between the camera and community. The density function becomes:

$$f_h(x) = \frac{1}{nh}\sum_{i=1}^{n} \max_{g \in G(x)} K(\frac{g - c_i}{h}), \tag{3}$$

where $G(x)$ denotes the set of all gates of community x.

To model the effect of *frequency*, we further add the weight of each camera into the KDE function of community. The weighted KDE is as follows:

$$f_h(x) = \frac{1}{nh}\sum_{i=1}^{n} w_{c_i} \max_{g \in G(x)} K(\frac{g - c_i}{h}), \tag{4}$$

where w_{c_i} is the number of times the vehicle passes through c_i.

We know that the cameras in the home area candidate are either first passing cameras (i.e., origin) or last passing cameras (i.e., destination) for the vehicle veh. Here we separately consider the effect of the origins and destinations on the community. Therefore, the final density function becomes:

$$f_h(x) = \frac{1}{|\Omega|h}\sum_{c_i\in\Omega} w_{c_i}\max_{g\in G(x)} K(\frac{g-c_i}{h}) + \frac{1}{|\Phi|h}\sum_{c_i\in\Phi} w_{c_i}\max_{g\in G(x)} K(\frac{g-c_i}{h}), \tag{5}$$

where Ω and Φ are the set of local cameras in the home area candidate as origin and destination respectively.

Namely, the probability that the vehicle belongs to the community is estimated by adding up with the density estimation of leaving the community and the density estimation of entering the community.

As the above discussion, the surveillance camera data is extremely noisy. So the remaining issue is how to choose the meaningful local cameras Ω and Φ for each community candidate, which we discuss in next section.

4.4.2. Local camera selection

As described in Section 4.3, for the discovered home area candidate, we have two categories of cameras: origin and destination. Let $\Omega = \{cam_o^1, cam_o^2, \ldots, cam_o^m\}$ be the set of origin cameras in the home area candidate, and $\Phi = \{cam_d^1, cam_d^2, \ldots, cam_d^n\}$ be the set of destination cameras in the home area candidate. Note that Ω and Φ may intersect since a camera may serve as either an origin or a destination for a vehicle. We now have a set of community candidates near the home area candidate, denoted by $CMT = \{cmt_1, cmt_2, \ldots, cmt_m\}$. In this section, our goal is to select the suitable local cameras from Ω and Φ for each community cmt_i in CMT to improve inference accuracy.

First, we choose the cameras within γ blocks as local cameras for each community, where a block refers to a road segment between two adjacent intersections. This is intuitive, because vehicles are rarely captured first by the cameras farther away if there are some cameras nearby. Even if they are captured, the number of times should be very small, and the distance is relatively far, thus this has little impact on the density function of community candidate.

Second, according to the direction of cameras and relative position of them and the nearby communities' gates, we further propose following two pruning rules to select the suitable local origin and destination cameras for each community.

Rule 1 (*Conflict Rule*)**.** *If a camera cam serves as an origin camera in the home area candidate, i.e., $cam \in \Omega$, but the direction of cam is conflict with all exit gates for the community cmt, then cam is pruned from Ω with respect to cmt.*

For example, in Fig. 3, camera c_2 is an origin camera (i.e., $c_2 \in \Omega$) for the home area candidate, however, c_2 is difficult to be an origin camera for the community cmt_1, because vehicles can first pass through c_2 and then reach at the community gates of cmt_1, and vehicles would not first pass through c_2 after going out from the community cmt_1. That is, c_2 should serve as a destination camera for community cmt_1. Therefore, we eliminate c_2 from Ω for community candidate cmt_1 in KDE function.

Similarly, we have another symmetric conflict rule: If a camera cam serves as a destination camera in the home area candidate, i.e., $cam \in \Phi$, but the direction of cam is conflict with all entrance gates for the community cmt, then cam is pruned from Φ with respect to cmt.

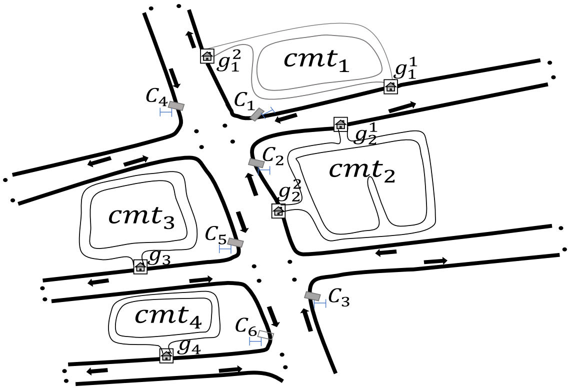


Fig. 3. A sample of local camera selection.

Rule 2 (*Sequence Rule*)**.** *Given an origin camera cam $\in \Omega$, if vehicles arrive at the camera cam from community cmt, they must need to pass another camera cam′ first, and if the number of times vehicle veh passes through cam′ is zero and cam′ works well during that time period, then cam is pruned from Ω for the community cmt.*

Again, we have another symmetric sequence rule: for a destination camera $cam \in \Phi$, if vehicles arrive at community cmt from the camera cam, they need to pass another camera cam' first, and if the number of times vehicle veh passes through cam' is zero and cam' works well during that time period, then cam is pruned from Φ for the community cmt.

For example, in Fig. 3, when a vehicle arrives at the gate of community cmt_4 from c_5, it has to pass through c_6 camera first. However, the number of visit to camera c_6 is zero in the home area candidate but c_6 captures other vehicles during same time period, so c_5 should be pruned for community cmt_4.

To implement the above pruning rules, we use the shortest path searching on the road network to obtain the path sequences between cameras and each community gate, consisting of passing road segments, intersections and cameras. Based on these path sequences, we decide whether the cameras meet the conditions of pruning.

It is worth noting that we use the road network distance between selected local cameras and community candidates. Unlike Euclidean distance, road network distance is directional, that is, vehicles run on the road along the road direction, and traverse the intersections and cameras in sequence, which better reflects the real travel distance between local cameras and community candidates.

4.5. Home location inference method

The pseudo-code for our **HomInf** (<u>Hom</u>e location <u>Inf</u>erence) framework is given in Algorithm 1.

First, we extract the *O-D* pairs from each vehicle trajectory Tr_{id} in the trajectory set TR for each vehicle (lines 1–7). Then we use DBSCAN clustering to get neighboring camera clusters (line 8). To distinguish the home area from other clusters, we further use *in/out* time pattern constraints to detect the home area candidate (lines 9–13). As shown in lines 14–15, we select the largest clusters with the maximum visit frequency, if there are multiple cluster satisfying *in/out* time pattern constraints.

Second, we divide the cameras in the home area candidate into Ω set (i.e., origin cameras) and Φ set (i.e., destination cameras) to model the effect of the origins and destinations separately (lines 16–20). Next, we select the local cameras in Ω and Φ set for

Algorithm 1 HomInf: Home Location Inference Algorithm

```
Input: Vehicle trajectory $TR$, road network $\mathcal{G}$, residential community set $CMT$
Output: Home community rank list $RL$
 1: for each $Tr_{id} \in TR$ do                              ▷ extract $O$-$D$ pairs.
 2:     for each $\langle cam_i, ts_i \rangle \in Tr_{id}$ do
 3:         if $|ts_{i+1} - ts_i| \geq \tau$ then
 4:             $O \leftarrow O \cup \{cam_{i+1}\}$;
 5:             $D \leftarrow D \cup \{cam_i\}$;
 6:     $O \leftarrow O \cup \{cam_{start}\}$;
 7:     $D \leftarrow D \cup \{cam_{end}\}$;
 8: $Clu \leftarrow$ DBSCAN($O, D, MinPts, Eps, \theta$);     ▷ use DBSCAN clustering $O$-$D$ pairs.
 9: for each $clu_i \in Clu$ do
10:     $In$ time pattern $\leftarrow clu_i \cap D$;
11:     $Out$ time pattern $\leftarrow clu_i \cap O$;
12:     if time matching then ▷ match with $in/out$ time patterns.
13:         $Can_{clu} \leftarrow Can_{clu} \cup \{clu_i\}$;
14: if $Can_{clu}.size > 1$ then                    ▷ select the largest cluster.
15:     $area \leftarrow \max(Can_{clu})$;
16: for each $cam_i \in area$ do
17:     if $cam_i \in O$ then
18:         $\Omega \leftarrow \Omega \cup \{cam_i\}$
19:     if $cam_i \in D$ then
20:         $\Phi \leftarrow \Phi \cup \{cam_i\}$
21: for each $cmt_i \in CMT(area)$ do
22:     $\Omega_i \leftarrow \Omega$;
23:     for $c_j \in \Omega_i$ do
24:         if $|c_j - cmt_i| \leq \gamma$ blocks then
25:             if $\forall g \in cmt_i,\ c_j \notin out(g)$ then
26:                 $\Omega_i \leftarrow \Omega_i / c_j$;           ▷ satisfy RULE 1
27:             if $\forall g \in cmt_i, \forall c \in path(g, c_j),\ c \notin \Omega_i$ and $c \in Cam$ then
28:                 $\Omega_i \leftarrow \Omega_i / c_j$;           ▷ satisfy RULE 2
29:         else
30:             $\Omega_i \leftarrow \Omega_i / c_j$;
31:     get $\Phi_i$ in the same way;
32:     $f(i) \leftarrow$ KDE($\Omega_i, \Phi_i$); ▷ compute community density using Eq. (5).
33: $RL \leftarrow$ Rank($CMT(area)$);
```

each community in the home area candidate. $CMT(area)$ in line 21 denotes all community candidates near/in the home area candidate. More specifically, we first get a copy of Ω for community cmt_i (line 22). We only retain the local cameras within γ blocks to community cmt_i (line 24 and lines 29–30). And we remove the cameras that satisfy Rules 1 and 2 (lines 25–28). Here $out(g)$ represents all cameras that vehicles would pass after getting out from gate g, and $path(g, c_j)$ denotes the set of cameras in the path from gate g to camera c_j. We next obtain Φ_i set for cmt_i in the same way. Then we compute the density of community cmt_i using selected local cameras Ω_i and Φ_i (line 32).

Finally, we obtain the inferred home community rank list RL by ranking computed density values of all communities (line 33).

5. Experiments

In this section, we conduct quantitative evaluations to demonstrate the effectiveness of the proposed framework on real-world datasets.

Table 2
Statistics of surveillance camera dataset.

Time span	08/01/2016-08/31/2016
Number of surveillance cameras	1704
Number of records	405,370,631
Number of total vehicles	11,299,927
Average number of vehicles per day	1,155,415

Table 3
Number of residential communities.

Districts	District1	District2	District3	District4	District5	total
#CMT	908	643	858	890	1105	4404

Table 4
Statistics of vehicles with ground truth.

Number of days	5–10	11–15	16–20	21–31	total
Number of vehicles	519	523	511	538	2091

5.1. Datasets

We use a real-world large-scale traffic dataset collected in a provincial capital of China and crawl the corresponding contextual datasets from multiple data sources.

- **Surveillance Camera Data.** The dataset contains 405, 370, 631 records from 1704 surveillance cameras over the period of Aug. 1st, 2016–Aug. 31st, 2016. Table 2 shows the statistics of the dataset in detail.
- **Road Network.** We crawl the road network of the city from the public-available OpenStreatMap [51]. The road network is comprised of 1034 intersections and 4350 road segments.
- **Residential Community Data.** We collect the residential community information using Baidu Map API.[1] The dataset includes all residential communities covering five most dense districts in the City. Table 3 shows the number of residential communities in each district.

5.2. Evaluation settings

5.2.1. Ground truth

In our study, we need to know the real home community of some vehicles as the ground truth. However, the real home locations of vehicles are not included in the surveillance camera data. To get the ground truth, we conduct a user study to collect the home community information in the provincial capital city. More specifically, we randomly investigate 3072 drivers from 12 communities on the spot in different districts in the morning or evening, and capture the records of their entry or exit to their home community. Therefore, we use the communities as the ground truth for the survey vehicles. Note that several selected communities are located in densely populated areas, meaning there are multiple residential communities around them. Since some vehicles were not captured in the surveillance camera data during Aug. 1st-Aug. 31st, 2016, we finally get 2091 vehicles with a real home community. Additionally, we make a statistics for the distribution of days captured in the surveillance camera data for these vehicles, as shown in Table 4.

5.2.2. Hyperparameter settings

We set the time interval threshold $\tau = 60$ minutes, which means that when the time interval between any two adjacent cameras in the vehicle trajectory exceeds 60 minutes, there is

[1] api.map.baidu.com/lbsapi/getpoint/index.html.

a stop point. We set frequency threshold $\theta = 3$, that is, we remove the cameras that each vehicle passes less than 3 times. We set block threshold $\gamma = 3$ for local camera selection. We set the bandwidth of $h = 1600$ for KDE model. If there is no real residential community in the inferred ranking of home communities, to calculate the *RMSE* and *MRR* values, we set the vehicle's real residential community ranking to 50.

5.3. Evaluation methods and metrics

5.3.1. Baselines

To the best of our knowledge, we are the first to infer home locations for vehicles with surveillance camera data. However, we compare our method with the following basic method and variations.

- **KNN.** KNN uses k nearest neighbors (cameras) to infer home community. We first extract and cluster the starting cameras and the ending cameras in vehicle trajectories, and select the largest cluster as the home area candidate for each vehicle. By calculating the distance between the community gate and each camera in the home location area, the k nearest cameras are selected to compute the distance sum for each residential community. The smaller the distance, the greater the probability that the vehicle belongs to the community. For the distance calculating, we use two kinds of distance measuring methods, one is Euclidean distance and the other is road network distance, denoted by **KNN-ED** and **KNN-RD** respectively.
- **HA-KNN. HA-KNN** first uses our proposed home area candidate discovery method to find the home area, and then adopts **KNN** method to infer the home community ranking list. Correspondingly, we use **HA-KNN-ED** and **HA-KNN-RD** to denote **HA-KNN** method using Euclidean distance and road network distance, respectively.

5.3.2. Metrics

We use the following four metrics to comprehensively evaluate the performance of the proposed method, which are defined as follow.

- *Top@n* **Accuracy** (*Top@n* **Acc**). *Top@n* evaluation method is to calculate the accuracy of the actual home community of each vehicle in the estimated *Top@n* communities. The calculation formula is as follows:

$$Top@n\ Acc = \frac{\#rank@n}{m}, \tag{6}$$

where m is the number of test vehicles, and $\#rank@n$ represents the number of vehicles in which the actual community is ranked within Top-n.

- **Root Mean Squared Error (RMSE).** We use *RMSE* to calculate the errors between the predicted top-1 community and the ranking of actual community for all tested vehicles. *RMSE* is defined as follows:

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}|rank_i - 1|^2}, \tag{7}$$

where $rank_i$ represents the ranking of the real residential community of the ith vehicle.

- **Mean Reciprocal Rank (MRR).** We use *MRR* to evaluate the performance of all methods. The larger *MRR* value represents the better performance. *MRR* is defined as follows:

$$MRR = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{rank_i}. \tag{8}$$

Table 5
Top@n accuracy results of all methods.

Top@n	*Top@1*	*Top@2*	*Top@3*	*Top@4*	*Top@5*
KNN-ED	0.2026	0.3281	0.3969	0.5160	0.6242
KNN-RD	0.2466	0.3767	0.4510	0.5747	0.6700
HA-KNN-ED	0.3025	0.4684	0.5509	0.6645	0.7534
HA-KNN-RD	0.3309	0.5124	0.5930	0.7131	0.7654
HomInf	**0.3868**	**0.5564**	**0.6517**	**0.7415**	**0.8121**

Table 6
MRR, RMSE and AED results of all methods.

Method	MRR	RMSE	AED
KNN-ED	0.3568	15.6098	1932
KNN-RD	0.3938	15.4673	1796
HA-KNN-ED	0.4783	13.2932	1359
HA-KNN-RD	0.5111	12.8917	1206
HomInf	**0.5547**	**11.2677**	**929**

- **Average Error Distance (AED).** We also use *AED* to calculate the distance error between the inferred top-1 community and the actual community for each vehicle. *AED* is defined as follows:

$$AED = \frac{1}{m}\sum_{i=1}^{m}\sqrt{dist(cmtf_i, cmta_i)^2}, \tag{9}$$

where $dist(cmtf_i, cmta_i)$ represents the Euclidean distance between the predicted top-1 community $cmtf_i$ and the actual home community $cmta_i$ for the ith vehicle.

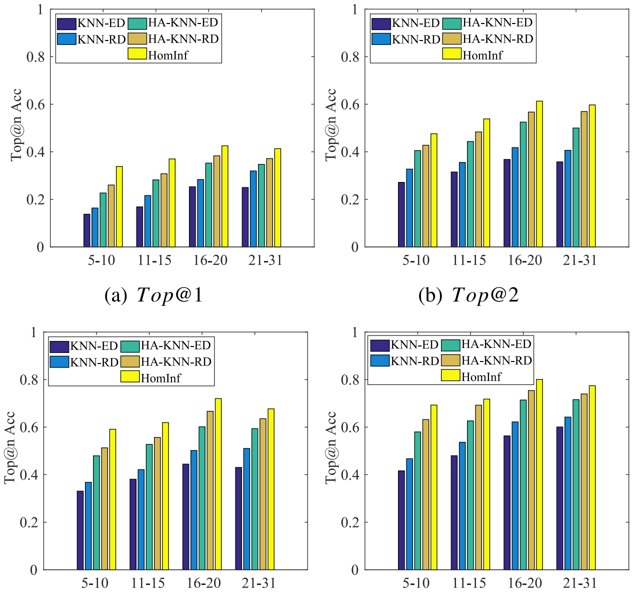
5.4. Overall performance

First, we evaluate the overall performance of our method and baselines in terms of *Top@n Acc*, *MRR*, *RMSE* and *AED* metrics. Table 5 shows the results of *Top@n Acc* for all methods, and Table 6 shows the *MRR*, *RMSE* and *AED* results of all methods.

From Table 5, we can see that our proposed **HomInf** significantly outperforms all baselines in all cases. As top-n increases, the accuracy of all methods increases. As we see, our **HomInf** method reaches at 81.21% accuracy when $top@n = 5$, which improves **KNN-ED** and **KNN-RD** by 30% and 21% in *Top@n Acc*, respectively. More specifically, the accuracy of our **HomInf** method is 38.68% when $top@n = 1$. It means that it is very difficult to directly infer the home community for vehicles in the surveillance camera system, because of the sparsity of surveillance cameras, high noise of the camera data, and multiple neighboring residential communities near some cameras. However, our method remarkably outperforms **KNN-ED** and **KNN-RD** by 91% and 57% improvements in the case of $top@n = 1$.

Top@n Acc focuses on the vehicles whose real communities are in the top-n of the inferred community rank-list. For example, when $top@n = 5$, for our **HomInf** method, we know that 81.21% vehicles' real communities are in the top-5 of the predicted community rank-list, but we do not know what the remaining 18.79% vehicles ranking are. Therefore, we also evaluate the performance of all methods in terms of *MRR*, *RMSE* and *AED*. In Table 6, we again see that **HomInf** is superior to all baselines in these three metrics. In particular, *MRR* value of our **HomInf** method is 0.5547, which improves **KNN** methods by 48% on average for all tested vehicles. And *AED* of **HomInf** method for all tested vehicles is 929 m, which is also much less than the average *AED* (1864 m) of **KNN** methods.

As shown in Tables 5 and 6, **HA-KNN** methods are better than corresponding **KNN** methods. This is because **HA-KNN** employs our proposed home area candidate discovery method, which effectively locate the most possible home area by maximally using vehicle trajectory records and time-aware constraints. This



(a) *Top*@1

(b) *Top*@2

(c) *Top*@3

(d) *Top*@4

Fig. 4. *Top@n ACC* results w.r.t. days.

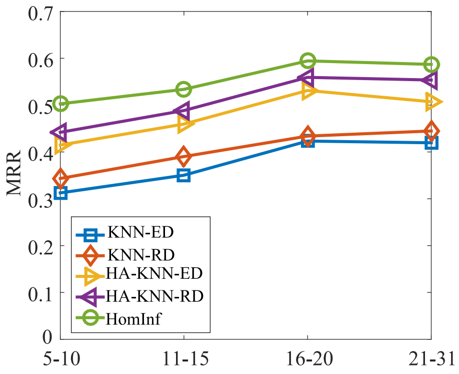


Fig. 5. MRR results w.r.t. days.

also demonstrates the effectiveness of our proposed home area candidate discovery method. Moreover, our **HomInf** further outperforms **HA-KNN** methods in all metrics, which illustrates the superiority of our proposed KDE-based model in predicting home community compared to KNN model.

We also observe that the methods that use road network distance outperform the counterparts that use Euclidean distance (i.e., **KNN-RD** vs. **KNN-ED**, and **HA-KNN-RD** vs. **HA-KNN-ED**). This indicates that the use of the road network distance is more effective than the Euclidean distance. Because vehicles run on the road network along the road direction and traverse the intersections and cameras in sequence, which better reflects the real travel distance between local cameras and community candidates compared to Euclidean distance.

5.5. *Performance w.r.t. data sparsity*

Next, we demonstrate that our method is also useful when the data is sparse. As shown in Table 4, we divide all tested vehicles into four groups according to the number of days captured in surveillance camera data: 5–10 days, 11–15 days, 16–20 days, and 21–31 days.

The performance of all methods w.r.t. the number of recorded days in terms of *Top@n ACC*, *MRR* and *AED* is shown in Figs. 4–6.

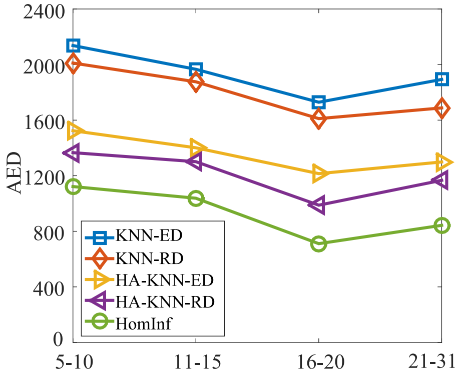


Fig. 6. AED results w.r.t. days.

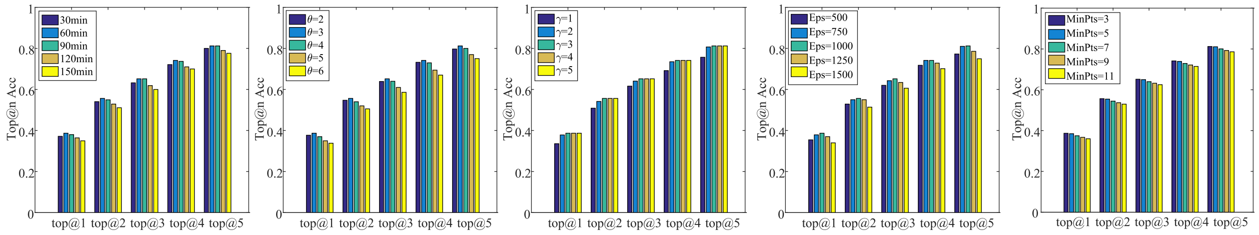
As we can see, our method is consistently better than all baseline methods on all groups. The first feeling tells us that the more data we have, the higher the accuracy of the prediction. However, we see that our **HomInf** method also performs very well with sparse data on the group of 5–10 days (Fig. 4). The vehicles in this group are the most difficult cases, where most of them only are captured in 5–7 days in surveillance camera data. Our method is much more effective for such challenging cases. *Top*@4 *ACC* of **KNN-ED** method is only 0.4164 for this group, while our **HomInf** achieves 0.6929 accuracy (Fig. 4(d)). This is because our method maximizes the use of vehicle trajectories in home area candidate discovery. We use *O-D* pair detection to make full use of each trajectory segments, which significantly alleviates the problem of data sparsity. Furthermore, we use *O-D* pair clustering and time-aware constraints to detect the home area more effectively. **HA-KNN** methods significantly outperforming **KNN** methods on this sparse group in terms of three metrics also just demonstrates this point.

Additionally, we also see that all methods including our method do not work better on the group of 21–31 days compared with the 16–20 group in all three metrics. This is because the surveillance camera data is extremely noise. Although the vehicle trajectory increases in the 21–31 group, a large number of noise cameras also appear in the discovered home area candidate. However, our **HomInf** method still performs very well and outperforms all baselines on this group. This is mainly because **HomInf** uses KDE method to model the effect of captured cameras on the home community with respect to distance and frequency. Especially, we propose the local camera selection method to choose the suitable camera for each community candidate, which effectively removes lots of conflicting noise.

5.6. *Parameter sensitivity*

We now investigate the performance of our method w.r.t. the important parameters in term of *Top@n ACC*.

Fig. 7(a) shows the accuracy of our **HomInf** method by varying time threshold τ from 30 to 150 minutes. As we can see, our method achieves the best performance when time threshold τ is between 60 and 90 minutes. When $\tau = 30$, the result is slightly worse. This is because more transitory stop points are detected when $\tau = 30$, which may result in an increase in the number of irrelevant cameras in home area candidate discovery and further affect the inference accuracy of home community. When $\tau \geq 120$, the accuracy result is getting worse. This is may because many important stays are not extracted. For example, some people go home from workplace at noon and then return to workplace in the early afternoon. But most of them have no more than 120



(a) Performance w.r.t. τ (b) Performance w.r.t. θ (c) Performance w.r.t. γ (d) Performance w.r.t. Eps (e) Performance w.r.t. $MinPts$

Fig. 7. Performance w.r.t. important parameters.

minutes' rest time. As a result, the meaningful local cameras in the home area may be omitted, resulting in a lower accuracy.

Fig. 7(b) shows the accuracy of our method w.r.t. the frequency threshold θ in *Top@n Acc*. We can see that the accuracy of our method achieves the best when $\theta = 3$ in all cases. When $\theta = 2$, more low-frequency cameras are added to the camera clusters in the home area candidate discovery, and the noise cameras may be introduced, which reduces the inference accuracy. However, the impact is smaller compared with the cases of $\theta = 5$ and $\theta = 6$. When $\theta \geq 5$, the accuracy of our method drops significantly, this is because many important cameras are removed in clustering, especially for the sparse cases. But for $\theta = 2$, although we introduce some incidental passing cameras in camera clusters, we can retain the important cameras and eliminate the conflicting cameras in the local camera selection. Therefore, the performance of our method for smaller θ value is better than that for larger θ value.

Fig. 7(c) shows the *Top@n Acc* of our method w.r.t. the block threshold γ. As expected, the performance of our method will no longer increase when γ reaches a certain size. Due to the sparseness of surveillance cameras, they may not cover all roads near residential communities, thus we need to utilize the farther cameras for inferring the home location. As described in Section 4.4.2, very far cameras also do not affect the density calculation of the home community. In Fig. 7(c), we observe that the accuracy of our method no longer changes when $\gamma \geq 3$, which means the important cameras have already been included for home location inference in our problem when using the adjacent three blocks.

Finally, we evaluate the performance of our method with respect to the clustering parameter *Eps* and *MinPts*. Fig. 7(d) shows the *Top@n Acc* results of our method by varying *Eps* from 500 to 1500 m. As we can see, our method achieves the best performance when *Eps* ranges from 750 to 1000 m. Namely, all *Eps* values chosen from 750 to 1000 are very suitable for our problem. This is because much more smaller camera clusters are generated by DBSCAN when $Eps = 500$, which may cause that wrong or split home area is selected in the home area candidate discovery. For example, the home area covering multiple further cameras may be split into two or more clusters when set a smaller *Eps*. When *Eps* value is set very larger, multiple frequently visited places may be merged into one cluster, which causes a decrease in the accuracy of home community inference in a much larger noisy area.

Fig. 7(e) shows the *Top@n Acc* results of our method by varying *MinPts* from 3 to 11. Since we omit the origin and destination cameras with low frequency that is less than θ (the default is 3) when performing DBSCAN clustering, experimental results are the same when $MinPts \leq 3$. As we can see, our method achieves the best performance when *MinPts* ranges from 3 to 5. The inference accuracy of the proposed method gradually decreases when $MinPts > 5$. This is because some camera clusters with low visiting frequency are neglected when performing DBSCAN with a larger *MinPts*. Especially for the sparse data (e.g., 5–10 days), the number of visits to the frequently visited places by a vehicle is inherently small. Therefore, the cluster that corresponds to the real home location may be ignored when *MinPts* is set to a larger value. But on the other hand, more camera clusters are obtained by DBSCAN for the relative dense data (e.g., 21–31 days) when *MinPts* is set to a smaller value. Namely, more frequently visited places are detected for a vehicle by DBSCAN with a small *MinPts* value. Nonetheless, our proposed home area candidate discovery method effectively distinguishes the home area from the obtained frequently visited places, which guarantees the inference accuracy of our proposed framework.

6. Conclusion and future work

In this paper, we propose a home location inference framework for vehicles, called **HomInf**, which effectively predicts the home community for each vehicle with surveillance camera data. Our framework mainly consists of three parts: data preprocessing, home area candidate discovery and KDE-based home community inference. First, we obtain a context-rich road network with residential communities and cameras by collecting and preprocessing multiple contextual data with surveillance camera data. Then, we effectively discover the home area candidate by clustering *O-D* pairs in vehicle trajectories and leveraging the proposed time-aware constraints. Finally, **HomInf** infers the home community by using KDE method to model the effect of passing cameras on the residential communities in the home area candidate. Experimental results on real-world large-scale data showed the effectiveness of our proposed method.

Surveillance camera data in urban traffic systems records the daily life scripts of people, which contains not only the location information of drivers' home community, but also many other interesting activities, such as workplace, and entertainments. In the future, we plan to predict the destinations for vehicles in such surveillance camera data by incorporating contextual POIs data.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Kai Chen: Methodology, Software, Data curation, Visualization, Writing - original draft. **Yanwei Yu:** Conceptualization, Supervision, Writing - review & editing, Funding acquisition. **Peng Song:** Data curation, Investigation. **Xianfeng Tang:** Validation, Writing - review & editing. **Lei Cao:** Writing - review & editing. **Xiangrong Tong:** Resources, Funding acquisition.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments and helpful suggestions. This work is partially supported by the National Natural Science Foundation of China under Grant Nos. : 61773331, 61703360, 61572418 and 61403328.

References

[1] Tatiana Pontes, Gabriel Magno, Marisa Vasconcelos, Aditi Gupta, Jussara Almeida, Ponnurangam Kumaraguru, Virgilio Almeida, Beware of what you share: Inferring home location in social networks, in: Data Mining Workshops (ICDMW), 2012 IEEE 12th International Conference on, IEEE, 2012, pp. 571–578.

[2] Kejiang Ren, Shaowu Zhang, Hongfei Lin, Where are you settling down: Geo-locating twitter users based on tweets and social networks, in: Asia Information Retrieval Symposium, Springer, 2012, pp. 150–161.

[3] Rui Li, Shengjie Wang, Hongbo Deng, Rui Wang, Kevin Chen-Chuan Chang, Towards social user profiling: unified and discriminative influence model for inferring home locations, in: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2012, pp. 1023–1031.

[4] Afshin Rahimi, Duy Vu, Trevor Cohn, Timothy Baldwin, Exploiting text and network context for geolocation of social media users, 2015, arXiv preprint arXiv:1506.04803.

[5] Yulong Gu, Yuan Yao, Weidong Liu, Jiaxing Song, We know where you are: Home location identification in location-based social networks, in: Computer Communication and Networks (ICCCN), 2016 25th International Conference on, IEEE, 2016, pp. 1–9.

[6] Jinpeng Chen, Yu Liu, Ming Zou, Home location profiling for users in social media, Inf. Manage. 53 (1) (2016) 135–143.

[7] Chao Huang, Dong Wang, Shenglong Zhu, Where are you from: Home location profiling of crowd sensors from noisy and sparse crowdsourcing data, in: INFOCOM 2017-IEEE Conference on Computer Communications, IEEE, IEEE, 2017, pp. 1–9.

[8] Axel Schulz, Aristotelis Hadjakos, Heiko Paulheim, Johannes Nachtwey, Max Mühlhäuser, A multi-indicator approach for geolocalization of tweets., in: Icwsm, 2013, pp. 573–582.

[9] Guoliang Li, Jun Hu, Jianhua Feng, Kian-lee Tan, Effective location identification from microblogs, in: Data Engineering (ICDE), 2014 IEEE 30th International Conference on, IEEE, 2014, pp. 880–891.

[10] Hariton Efstathiades, Demetris Antoniades, George Pallis, Marios D Dikaiakos, Identification of key locations based on online social network activity, in: Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015, ACM, 2015, pp. 218–225.

[11] Tian-ran Hu, Jie-bo Luo, Henry Kautz, Adam Sadilek, Home location inference from sparse and noisy data: models and applications, Front. Inf. Technol. Electron. Eng. 17 (5) (2016) 389–402.

[12] Nabil Hossain, Tianran Hu, Roghayeh Feizi, Ann Marie White, Jiebo Luo, Henry Kautz, Inferring fine-grained details on user activities and home location from social media: Detecting drinking-while-tweeting patterns in communities, 2016, arXiv preprint arXiv:1603.03181.

[13] Adam Poulston, Mark Stevenson, Kalina Bontcheva, Hyperlocal home location identification of twitter profiles, in: Proceedings of the 28th ACM Conference on Hypertext and Social Media, ACM, 2017, pp. 45–54.

[14] Jie Lin, Robert G. Cromley, Inferring the home locations of twitter users based on the spatiotemporal clustering of twitter data, Trans. GIS 22 (1) (2018) 82–97.

[15] John Krumm, Inference attacks on location tracks, in: International Conference on Pervasive Computing, Springer, 2007, pp. 127–143.

[16] Xin Cao, Gao Cong, Christian S. Jensen, Mining significant semantic locations from gps data, Proc. VLDB Endow. 3 (1–2) (2010) 1009–1020.

[17] Chengcheng Wan, Yanmin Zhu, Jiadi Yu, Yanyan Shen, Smopat: Mining semantic mobility patterns from trajectories of private vehicles, Inform. Sci. 429 (2018) 12–25.

[18] Fei Wu, Hongjian Wang, Zhenhui Li, Semantic exploration of traffic dynamics, 2018, arXiv preprint arXiv:1804.04165.

[19] Swarup Chandra, Latifur Khan, Fahad Bin Muhaya, Estimating twitter user location using social interactions–a content based approach, in: Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on, IEEE, 2011, pp. 838–843.

[20] Brent Hecht, Lichan Hong, Bongwon Suh, Ed H. Chi, Tweets from justin bieber's heart: the dynamics of the location field in user profiles, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, 2011, pp. 237–246.

[21] Bo Han, Paul Cook, Timothy Baldwin, A stacking-based approach to twitter user geolocation prediction, in: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations, 2013, pp. 7–12.

[22] Zhiyuan Cheng, James Caverlee, Kyumin Lee, You are where you tweet: a content-based approach to geo-locating twitter users, in: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, ACM, 2010, pp. 759–768.

[23] KyoungMin Ryoo, Sue Moon, Inferring twitter user locations with 10 km accuracy, in: Proceedings of the 23rd International Conference on World Wide Web, ACM, 2014, pp. 643–648.

[24] Jalal Mahmud, Jeffrey Nichols, Clemens Drews, Home location identification of twitter users, ACM Trans. Intell. Syst. Technol. (TIST) 5 (3) (2014) 47.

[25] Yuki Kondo, Masatsugu Hangyo, Mitsuo Yoshida, Kyoji Umemura, Home location estimation using weather observation data, in: Advanced Informatics, Concepts, Theory, and Applications (ICAICTA), 2017 International Conference on, IEEE, 2017, pp. 1–6.

[26] Yasuhide Miura, Motoki Taniguchi, Tomoki Taniguchi, Tomoko Ohkuma, Unifying text, metadata, and user network representations with a neural network for geolocation prediction, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vol. 1, 2017, pp. 1260–1272.

[27] Yu Zheng, Lizhu Zhang, Xing Xie, Wei-Ying Ma, Mining interesting locations and travel sequences from gps trajectories, in: Proceedings of the 18th International Conference on World Wide Web, ACM, 2009, pp. 791–800.

[28] Fei Wu, Zhenhui Li, Wang-Chien Lee, Hongjian Wang, Zhuojie Huang, Semantic annotation of mobility data using social media, in: Proceedings of the 24th International Conference on World Wide Web, International World Wide Web Conferences Steering Committee, 2015, pp. 1253–1263.

[29] Fei Wu, Hongjian Wang, Zhenhui Li, Interpreting traffic dynamics using ubiquitous urban data, in: Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, ACM, 2016, p. 69.

[30] Jalal Mahmud, Jeffrey Nichols, Clemens Drews, Where is this tweet from? inferring home locations of twitter users., ICWSM 12 (2012) 511–514, 2012.

[31] Yasuhide Miura, Motoki Taniguchi, Tomoki Taniguchi, Tomoko Ohkuma, A simple scalable neural networks based model for geolocation prediction in Twitter, in: Proceedings of the 2nd Workshop on Noisy User-Generated Text (WNUT), 2016, pp. 235–239.

[32] Hamdi Kavak, Daniele Vernon-Bido, Jose J. Padilla, Fine-scale prediction of people's home location using social media footprints, in: International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation, Springer, 2018, pp. 183–189.

[33] Tatiana Pontes, Marisa Vasconcelos, Jussara Almeida, Ponnurangam Kumaraguru, Virgilio Almeida, We know where you live: privacy characterization of foursquare behavior, in: Proceedings of the 2012 ACM Conference on Ubiquitous Computing, ACM, 2012, pp. 898–905.

[34] Jinpeng Chen, Yu Liu, Ming Zou, From tie strength to function: Home location estimation in social network, in: Computing, Communications and IT Applications Conference (ComComAp), 2014 IEEE, IEEE, 2014, pp. 67–71.

[35] Lars Backstrom, Eric Sun, Cameron Marlow, Find me if you can: improving geographical prediction with social and spatial proximity, in: Proceedings of the 19th International Conference on World Wide Web, ACM, 2010, pp. 61–70.

[36] David Jurgens, That's what friends are for: Inferring location in online social media platforms based on social relationships., Icwsm 13 (13) (2013) 273–282.

[37] Afshin Rahimi, Trevor Cohn, Timothy Baldwin, Twitter user geolocation using a unified text and network prediction model, 2015, arXiv preprint arXiv:1506.08259.

[38] Hicham G. Elmongui, Hader Morsy, Riham Mansour, Inference models for twitter user's home location prediction, in: Computer Systems and Applications (AICCSA), 2015 IEEE/ACS 12th International Conference of, IEEE, 2015, pp. 1–8.

[39] Yulong Gu, Jiaxing Song, Weidong Liu, Lixin Zou, Hlgps: A home location global positioning system in location-based social networks, in: Data Mining (ICDM), 2016 IEEE 16th International Conference on, IEEE, 2016, pp. 901–906.

[40] Jeffrey McGee, James Caverlee, Zhiyuan Cheng, Location prediction in social media based on tie strength, in: Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, ACM, 2013, pp. 459–468.

[41] Longbo Kong, Zhi Liu, Yan Huang, Spot: Locating social media users based on social network context, Proc. VLDB Endow. 7 (13) (2014) 1681–1684.

[42] Chao Huang, Dong Wang, Jun Tao, An unsupervised approach to inferring the localness of people using incomplete geotemporal online check-in data, ACM Trans. Intell. Syst. Technol. (TIST) 8 (6) (2017) 80.

[43] Yu Zheng, Quannan Li, Yukun Chen, Xing Xie, Wei-Ying Ma, Understanding mobility based on gps data, in: Proceedings of the 10th International Conference on Ubiquitous Computing, ACM, 2008, pp. 312–321.

[44] Xiangye Xiao, Yu Zheng, Qiong Luo, Xing Xie, Finding similar users using category-based location history, in: Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, ACM, 2010, pp. 442–445.

[45] Jiangchuan Zheng, Lionel M. Ni, An unsupervised framework for sensing individual and cluster behavior patterns from human mobile data, in: Proceedings of the 2012 ACM Conference on Ubiquitous Computing, ACM, 2012, pp. 153–162.

[46] Quan Yuan, Gao Cong, Zongyang Ma, Aixin Sun, Nadia Magnenat Thalmann, Who, where, when and what: discover spatio-temporal topics for twitter users, in: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2013, pp. 605–613.

[47] John Krumm, Dany Rouhana, Placer: semantic place labels from diary data, in: Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing, ACM, 2013, pp. 163–172.

[48] Tianran Hu, Decode human life from social media, in: 2018 ACM Multimedia Conference on Multimedia Conference, ACM, 2018, pp. 820–824.

[49] Zhenhui Li, Jingjing Wang, Jiawei Han, Mining event periodicity from incomplete observations, in: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2012, pp. 444–452.

[50] Zhenhui Li, Jingjing Wang, Jiawei Han, Eperiodicity: Mining event periodicity from incomplete observations, IEEE Trans. Knowl. Data Eng. 27 (5) (2015) 1219–1232.

[51] Mordechai Haklay, Patrick Weber, Openstreetmap: User-generated street maps, IEEE Pervas Comput. 7 (4) (2008) 12–18.