Label-Based Trajectory Clustering in Complex Road Networks

Xinzheng Niu, Ting Chen, Chase Q. Wu, Jiajun Niu, and Yuran Li

***Abstract*— In the data mining of road networks, trajectory clustering of moving objects is of particular interest for its practical importance in many applications. Most of the existing approaches to this problem are based on distance measurement, and suffer from several performance limitations including inaccurate clustering, expensive computation, and incompetency to handle high dimensional trajectory data. This paper investigates the complex network theory and explores its application to trajectory clustering in road networks to address these issues. Specifically, we model a road network as a dual graph, which facilitates an effective transformation of the clustering problem from sub-trajectories in the road network to nodes in the complex network. Based on this model, we design a label-based trajectory clustering algorithm, referred to as LBTC, to capture and characterize the essence of similarity between nodes. For the evaluation of clustering performance, we establish a clustering criterion based on the classical Davies-Bouldin Index (DB), Akaike Information Criterion (AIC), and Bayesian Information Criterion (BIC) to maximize inter-cluster separation and intra-cluster homogeneity. The clustering accuracy and performance superiority of the proposed algorithm are illustrated by extensive simulations on both synthetic and real-world dataset in comparison with existing algorithms.**

***Index Terms*— Vehicle trajectory, road networks, clustering, dual graph.**

I. Introduction

THE growing use of GPS receivers and WiFi-embedded mobile devices equipped with storage hardware enables the collection of an enormous amount of trajectory data on route. Such data can be used for knowledge discovery in support of many applications for various purposes, for example, (1) determine the distribution of hot spots for tourist attraction recommendation and location-based precision advertising by merchants; (2) identify various zones with different levels of congestion to facilitate urban construction management (e.g., site selection for road expansion, coordination of urban spatial distribution, etc.), traffic control management (e.g., balancing traffic loads in road networks, guiding and adjusting traffic flows, etc.), and commercial outlet planning. Many of these applications require clustering the trajectories of mobile objects (mainly vehicles) in spatially constrained road networks.

Existing algorithms for vehicle trajectory clustering largely rely on distance measurement for either whole trajectory clustering or sub-trajectory clustering, and a large amount of work has been devoted to defining and measuring appropriate trajectory distances. For example, the work in [1]–[3], and [4] proposed trajectory clustering approaches in road networks based on various distance metrics. Traditional clustering methods using Euclidean distance oftentimes lead to inaccurate clustering mainly because trajectories may be of different lengths [5]. To address this issue, several methods have been proposed based on warping distance [6]–[9]. Another line of research attempts to extract and identify the geometric features of different trajectories, in particular their shapes. Shape-based distance such as Hausdorff [10] and Frechet [11] distance could be adapted to trajectories but has not been used for comparison as a single entity [5]. Lin and Su [12] proposed a method based exclusively on the trajectory shape but at a high computational cost. Moreover, vehicles may frequently change lanes while in motion, making it very challenging to calculate an accurate distance of any type between pairwise trajectories.

In recent years, the complex network theory has been extensively studied and applied for traffic analysis in Geographical Information System (GIS) to analyze the complexity and functionality of network structures. Since road networks constitute the main infrastructure of the transportation system in a city, the layout of the entire city would reflect the distribution of various geographical elements, featuring the structural characteristics of a complex network system. There exist two types of abstract models for road traffic networks, i.e., original graph and dual graph. To maintain the city's overall shape, one may employ the dual graph method to transform road segments and road intersections into nodes and edges, respectively, to construct the dual graph of a road network based on the generalized network topology [13].

Considering the limitations of existing distance-based clustering approaches, we propose a label-based trajectory clustering algorithm, referred to as LBTC, in road networks based on the complex network theory. LBTC leverages the key idea of Label Propagation Algorithm (LPA) [14], which is a fast graph-based semi-supervised learning algorithm for

Manuscript received February 22, 2018; revised August 2, 2018, March 24, 2019, and August 3, 2019; accepted August 7, 2019. This work was supported in part by the Scientific Research Project of Sichuan Provincial Public Security Department under Grant 2015SCYYCX06, in part by the Science and Technology Planning Project of Sichuan Province under Grant 2017FZ0094, and in part by the Fundamental Research Funds for the Central Universities Project with the University of Electronic Science and Technology of China. The Associate Editor for this article was S. A. Birrell. *(Corresponding author: Xinzheng Niu.)*

X. Niu, T. Chen, J. Niu, and Y. Li are with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: xinzhengniu@uestc.edu.cn).

C. Q. Wu is with the Department of Computer Science, New Jersey Institute of Technology, Newark, NJ 07102 USA (e-mail: chase.wu@njit.edu).

Digital Object Identifier 10.1109/TITS.2019.2937882

community detection with a near linear time complexity. One salient feature of the proposed LBTC algorithm that combines both the complex network theory and the community detection algorithm is that it obviates the need to compute the distance between trajectories, which is the main cause of the limitations of many existing distance-based approaches. Specifically, we first partition the given trajectories into a set of sub-trajectories. Then, we organize these sub-trajectories associated with the same road segment into an initial local dense cluster, since the objects moving along the same road segment have similar movement behaviors under the spatial constraints of the road network. Next, we convert the road network with such initial clusters into a dual graph topology and leverage the conventional label propagation algorithm for trajectory clustering, which is mainly adapted in the design of label update criteria to take into account the spatial constraints, such as traffic flows on consecutive road segments, road segment density as well as movement behaviors of mobile objects. For the evaluation of clustering performance, we establish clustering criteria based on the classical Davies-Bouldin index [15], Akaike Information Criterion [16], and Bayesian Information Criterion [17] to maximize inter-cluster separation and intra-cluster homogeneity. The clustering accuracy and performance superiority of the proposed algorithm are illustrated by extensive simulations on both synthetic and real-world dataset in comparison with existing algorithms that are widely used in practice.

The contributions of our work are summarized as follows:

- A transformation process that converts a series of sub-trajectories associated with the same road segment in the road network to a node in the complex network and constructs a corresponding topology graph.
- A label-based trajectory clustering algorithm with a novel update criterion that considers spatial constraints of road networks to obviate the use of distance computation between trajectories.
- A Process-Transformation-Clustering framework that provides a trajectory clustering analysis model with Davies-Bouldin (DB) Index, Akaike Information Criterion (AIC), and Bayesian Information Criterion (BIC) for clustering performance evaluation.

As illustrated in Fig. 1, the proposed LBTC approach consists of three main components: i) raw trajectory processing to partition each trajectory into a series of sub-trajectories according to the intersection distribution of road segments; ii) transformation to convert each road segment to a node using the dual graph method; and iii) clustering to group nodes using the label propagation algorithm. These technical components are detailed in Section IV. Compared with the traditional clustering approaches based on distance computation, our label-based clustering approach achieves higher accuracy and better performance.

The rest of the paper is organized as follows. Section II provides a survey of related work. Section III presents the cost models and provides a rigorous definition of the trajectory clustering problem. The design details of LBTC are provided in Section IV. We evaluate the accuracy and efficiency of LBTC in Section V and conclude our work in Section VI.

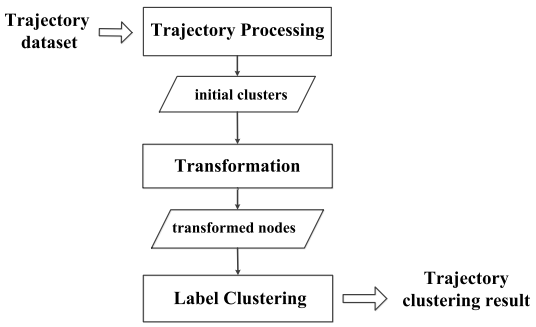


Fig. 1. Overview of our proposed LBTC.

II. Related Work

In recent years, the clustering analysis of mobile object trajectories in road networks has attracted a great deal of attention for various purposes such as understanding and exploring potential social connections and common interests of mobile users moving in a road network [3]. As a result, many techniques have been proposed in the literature to improve the accuracy and efficiency of vehicle trajectory clustering. We provide below a brief survey of such related work, including community detection used by the proposed LBTC solution for trajectory clustering.

Most existing efforts in trajectory clustering are focused on distance-based clustering of vehicle trajectories to establish similarities between trajectories. There are three main distance metrics used by various techniques in the literature, i.e., Euclidean distance, shape-based distance, and warping-based distance.

Euclidean distance is the most frequently used distance metric, which compares discrete objects of the same length. Trajectory-OPTICS [18], which extends OPTICS algorithm [19], is a good example for grouping similar trajectories as a whole, where the distance between two trajectories is the average Euclidean distance between two objects at every time stamp. TraClus [20], one of the most cited work on trajectory clustering that aims to find similar sub-trajectories rather than whole trajectories, clusters sub-trajectories based on Euclidean distance and then performs a DBSCAN-like [21] clustering by designing a distance function to define the density of sub-trajectories.

Euclidean distance-based clustering methods suffer from clustering inaccuracy mainly because trajectories could be of different lengths [5]. When the utility of spatial clustering of mobile object trajectories is considered in road network-aware location-based applications, density and Euclidean distance are no longer the effective measures. Hence, several methods using shape-based distance have been proposed to extract and identify the geometric features of trajectories. Hausdorff distance and Frechet distance are among the most well known shape-based distances. NEAT [22] addresses this problem by modifying the popular Hausdorff distance function to measure the distance between two flow clusters, and the proximity

measure in a road network space uses the shortest path distance instead of Euclidean distance. Shape-based distances could be adapted to trajectories, but may not be suitable for comparison as a single entity.

Some other clustering approaches use warping-based distance, which takes into account the temporal dimension. Several warping-based distances have been defined and used in the literature, such as Dynamic Time Warping (DTW) [6], Longest Common SubSequence (LCSS) [23], and Edit Distance on Real Sequences (EDR) [9]. The work in [24] achieved a comparable level of accuracy as the state-of-the-art K-Means clustering method using DTW and is more scalable to datasets with longer trajectories. Also, the work in [23] demonstrated that the LCSS distance is the most accurate and efficient for clustering even in the case of different sampling rates and noises.

The aforementioned approaches rely on trajectory distance computation to measure the similarity between whole trajectories or sub-trajectories. However, distance computation has caused several drawbacks such as inaccurate clustering and high computational cost, which calls for the design of clustering algorithms without distance computation. Typical examples include model-based methods [25], [26] and non-parametric hierarchical Bayesian methods [27], [28]. The former, as exemplified by Gaussian Mixture Models (GMM) [29] and Polynomial Mixture Regression (PMR) [30], requires a predefined cluster count; while the latter, as exemplified by Hierarchical Dirichlet Process (HDP) [31], Dual Hierarchical Dirichlet Process (Dual-HDP) [32], [33] and Stochastic Variational Dual Hierarchical Dirichlet Process (SV-DHDP) [34], does not require a predefined cluster count, but needs to set concentration parameters and base probability measures. In general, as the hierarchical level increases, hierarchical Bayesian models become more insensitive to the choice of these hyperparameters, and choosing an appropriate parameter can be non-trivial. Moreover, in probability distribution-based algorithms, one attempts to reproduce the observed realization of data points as a mix of predefined probability distribution functions, and the accuracy of such methods depends on the capability of the trial probability to represent the data [35]. Therefore, these methods cannot be directly employed to solve our problem.

Many systems in practice can be modeled as complex networks with several common characteristics such as small world, scale-free, and modular structure (also called community structure) [36]. Typical examples include power networks, airline networks, transportation networks, computer networks, and social networks, where the nodes tend to exhibit certain cluster characteristics. Community detection is a technique aimed at revealing the aggregated behaviors of complex networks. In general, community detection is considered as one type of clustering based on the graph structure in complex networks. Label Propagation Algorithm (LPA) was first proposed by Raghavan *et al.* [14] in 2007 and has gained its popularity for community detection because of the effectiveness and low complexity. Our research is largely inspired by these existing efforts to apply the complex network theory to trajectory clustering. To the best of our knowledge, we are among the first to cluster road trajectories in combination with community detection.

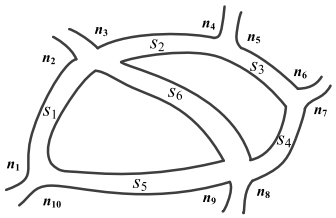


Fig. 2. A simple road network.

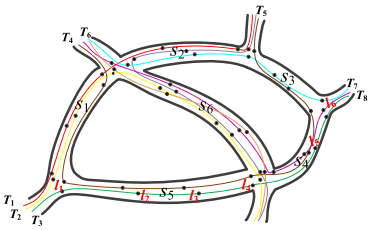


Fig. 3. An example of road trajectory.

III. Cost Models and Problem Definition

A. Road Network and Trajectory Processing

A road network consists of a set of intersections or junctions $\{n_1, n_2, \ldots, n_{N_n}\}$ and a set of road segments $\{s_1, s_2, \ldots, s_{N_s}\}$, each of which is denoted as $s_{sid} = (n_{start}, n_{end})$, $sid = 1, 2, \ldots, N_s$, where n_{start} and n_{end} are two intersections of the road network, representing the start point and the end point of the road segment, respectively. Fig. 2 shows a simple road network, comprised of a set of ten intersection points $\{n_1, n_2, \ldots, n_{10}\}$,and a set of six road segments $\{s_1, s_2, \ldots, s_6\}$, where $s_4 = (n_7, n_8)$.

We define a trajectory as a sequence of positions of a moving object in the road network over a continuous period of time, denoted by $T_{id} = \{l_1, l_2, \ldots, l_n\}$, where id is a unique identity of the trajectory, and $l = (sid, (x, y))$ denotes a geographical position with the coordinates of (x, y) in the sid road segment where the moving object travels. Moreover, according to the distribution of road segments and intersections, we define a sub-trajectory t_i as a set of $m + 1$ consecutive points on T_{id}, i.e., $t_i = \{l_f, l_{f+1}, \ldots, l_{f+m}\}$, $1 \leq f < f+m \leq n$. Thus, trajectory T_{id} can also be denoted as $T_{id} = \{t_1, t_2, \ldots, t_{N_t}\}$, $N_t < n$. As illustrated in Fig. 3, there are eight objects traveling in the road network of Fig. 2, marked in different colors, and six points are collected for T_3, i.e., $T_3 = \{l_1, l_2, \ldots, l_6\}$, which can be divided into two sub-trajectories $\{l_1, l_2, l_3, l_4\}$ and $\{l_5, l_6\}$, according to the intersection information.

Since the objects moving along the same road segment have similar movement behaviors under the spatial constraints of the road network, their trajectories should be grouped together regardless of the differences in their specific movements. Hence, it is unnecessary to further partition these trajectories, even though some rapid changes in the moving

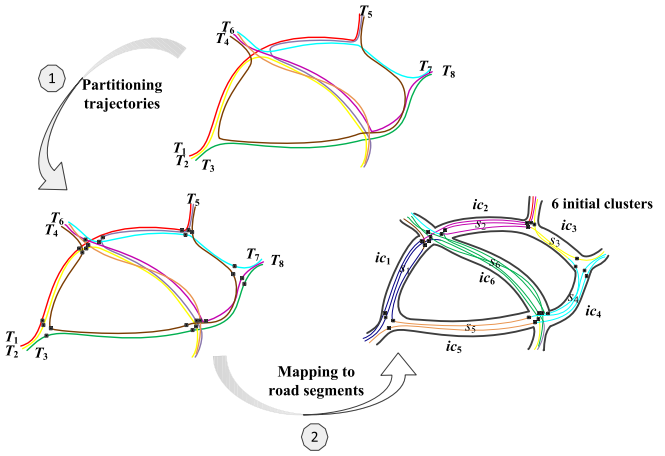


Fig. 4. Trajectory partitioning and mapping to road segments.

process may occur. Specifically, we take every two consecutive locations l_i and l_{i+1} in the trajectory being examined, and check if their road segment identifiers, denoted by $sid(l_i)$ and $sid(l_{i+1})$, respectively, are different. If $sid(l_i) = sid(l_{i+1})$, it indicates that both of them belong to the same road segment; otherwise, l_i and l_{i+1} are in different road segments and should be assigned to their respective road segments. Moreover, if l_i and l_{i+1} are contiguous, we need to obtain the road junction node that intersects these two road segments using the Selective Look-Ahead Map Matching (SLAMM) algorithm [37], a map-matching of an object location to the road network, and insert these obtained junction nodes into l_i and l_{i+1} as new points, which serve as the trajectory splitting points to partition the trajectory into sub-trajectories. Therefore, each trajectory $T = \{l_1, l_2, \ldots, l_n\}$ is divided into a set of sub-trajectories $\{t_1, t_2, \ldots t_{N_t}\}$, $N_t < n$, from its start position l_1 to the end position l_n, corresponding to the intersections of the road segments.

After all trajectories have been partitioned, the sub-trajectories associated with the same road segment are assembled together, forming a set of initial clusters, denoted by ic. An initial cluster is a group of distinct sub-trajectories, each of which belongs to the same road segment. This one-to-one mapping relationship between an initial cluster and its corresponding road segment plays a crucial role in the transformation process. Fig. 4 illustrates the process of trajectory partitioning and mapping to road segments, resulting in a set of initial clusters $\{ic_1, ic_2, \ldots, ic_5\}$ corresponding to $\{s_1, s_2, \ldots, s_5\}$.

B. Transformation

Connectivity and geometric features are the most basic and important attributes of road networks, and are also the key factors that determine the overall shape and affect the function of road networks. In recent years, the complex network theory has been increasingly applied to traffic analysis in GIS to understand the complexity and improve the functional characteristics of the road network structure.

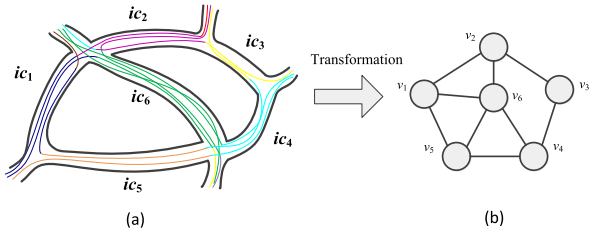


Fig. 5. An example of road network dual graph.

In complex network analysis, there exist two methods to construct network topology: original graph and dual graph. In the former, a road intersection is abstracted as a node and a road segment is abstracted as an edge; while in the latter, each node represents a road segment, and the connection between two nodes represents the connectivity of the two corresponding road segments in the road network. Compared with the original graph, the dual graph retains the overall network shape, while reflecting the traffic information of the road network.

As explained in Section III-A, there is a one-to-one mapping relationship between an initial cluster and its corresponding road segment, which enables us to employ the dual graph method to describe the topological structure of the road network, where each initial cluster is represented by a node v. For illustration, Fig. 5 (a) plots the original road network with initial clusters, and Fig. 5 (b) shows the dual graph topology, where initial clusters $\{ic_1, ic_2, \ldots, ic_6\}$ in the road network space are transformed into a set of nodes $\{v_1, \ldots, v_6\}$ in the Euclidean space while preserving their relative distances and relationships in the transformed space.

We provide the following definitions to facilitate the formulation of the trajectory clustering problem under study.

Definition 1: Neighbor Node: the neighbor nodes of v_i, denoted by $\pi(v_i)$, are a set of nodes directly connected to v_i, and the degree of v_i is the number $|\pi(v_i)|$ of neighbor nodes.

There are three well known variables in the traditional theory for uninterrupted traffic flow, i.e., density, flow, and speed. Thus, node v automatically inherits the attributes of road segments, i.e., initial clusters.

Definition 2: Node Density: the density of node v_i, denoted by $density(v_i)$, is the number of sub-trajectories in initial cluster ic_i corresponding to node v_i.

Note that node density directly reflects the importance of nodes: a higher density indicates a higher importance.

Definition 3: Node Flow: the flow between two nodes v_i and v_j, denoted by $flow(v_i, v_j)$, is the number of sub-trajectories contained in both initial cluster ic_i and ic_j corresponding to v_i and v_j, respectively, i.e., the number of common objects traveled on both road segments s_i and s_i.

Definition 4: Node Speed Limit: the speed limit of node v_i, denoted by $speed(v_i)$, is the speed limit of road segment s_i corresponding to node v_i.

Definition 5: Structural Similarity: given a dual graph topology, the structural similarity of nodes v_i and v_j, denoted by $SS(v_i, v_j)$, is measured by their degree and the number of

shared neighbor nodes as the Salton index (also called cosine similarity) [38], calculated as follows:

$$SS(v_i, v_j) = \frac{|\pi(v_i) \cap \pi(v_j)|}{\sqrt{|\pi(v_i)| \times |\pi(v_j)|}}. \quad (1)$$

The cosine similarity can be used to measure the likelihood of two nodes being in the same cluster.

Definition 6: Flow Factor: given a pair of nodes v_i and v_j, where v_i is connected to v_j directly, namely, $v_j \in \pi(v_i)$, the relative flow between v_i and v_j, denoted by $\delta(v_i, v_j)$, is computed from $density(v_i)$ and $flow(v_i, v_j)$ as follows:

$$\delta(v_i, v_j) = \frac{flow(v_i, v_j)}{density(v_i)}. \quad (2)$$

Note that $\delta(v_i, v_j)$ differs from $\delta(v_j, v_i)$, as $density(v_i) \neq density(v_j)$.

Definition 7: Speed Limit Factor: for node v_i, the relative speed between the moving objects of neighbor node $v_j \in \pi(v_i)$, denoted by $\beta(v_i, v_j)$, is computed from the speed of v_j and $\pi(v_i)$ as follows:

$$\beta(v_i, v_j) = \frac{speed(v_j)}{\sum_{v \in \pi(v_i)} speed(v)}. \quad (3)$$

Note that $\beta(v_i, v_j)$ differs from $\beta(v_j, v_i)$, as $speed(v_i) \neq speed(v_j)$ and $\sum_{v \in \pi(v_i)} speed(v) \neq \sum_{v \in \pi(v_j)} speed(v)$.

Definition 8: Attribute Similarity: the attribute similarity $AS(v_i, v_j)$ between nodes v_i and v_j is measured by flow factor $\delta(v_i, v_j)$ and speed limit factor $\beta(v_i, v_j)$ as follows:

$$AS(v_i, v_j) = \omega_\delta \cdot \delta(v_i, v_j) + \omega_\beta \cdot \beta(v_i, v_j). \quad (4)$$

where ω_δ and ω_β are the weights of δ and β, respectively, which satisfy $\omega_\delta \geq 0, \omega_\beta \geq 0, \omega_\delta + \omega_\beta = 1$. Obviously, $AS(v_i, v_j) \neq AS(v_j, v_i)$.

The weights ω_δ and ω_β are usually dependent on the application under consideration. For example, if the application supports both of these two factors at the same time, then we set $\omega_\delta = \omega_\beta = 1/2$. If the application emphasizes the flowing property of traffic, we may set $(\omega_\delta, \omega_\beta) = (1, 0)$. If the goal is to find out the route with the fastest speed, we may set $(\omega_\delta, \omega_\beta) = (0, 1)$. In our work, we set these weights equally by default, which works well in many applications and data sets.

C. Problem Formulation

We consider a set of trajectories $\mathcal{T} = \{T_1, T_2, \ldots, T_{N_\mathcal{T}}\}$ to be divided into a number of clusters $\mathcal{C} = \{C_1, C_2, \ldots, C_{N_\mathcal{C}}\}$. The quality of clustering is usually evaluated by inter-cluster separation and intra-cluster homogeneity [39]. In general, a larger inter-cluster separation and a higher degree of intra-cluster homogeneity indicates a more accurate clustering. In this work, we adopt Davies-Bouldin (DB) Index [15], Akaike Information Criterion (AIC) [16], and Bayesian Information Criterion (BIC) [17], which are widely used for cluster validation, to measure the clustering quality in road networks.

We first define the compactness measure $\overline{CP_i}$ [40], [41] of cluster C_i as follows:

$$\overline{CP_i} = \frac{1}{|C_i|} \sum_{v_i \in C_i} \|v_i - \mu_{C_i}\|_2, \quad (5)$$

where $|C_i|$ is the number of nodes in cluster C_i, and $\|v_i - \mu_{C_i}\|_2$ measures the Euclidean distance from node v_i to its assigned cluster centroid μ_{C_i}. In this work, the node with the greatest density in cluster C_i is considered as the cluster centroid μ_{C_i}.

The DB Index is defined as

$$DB = \frac{1}{N_\mathcal{C}} \sum_{i=1}^{N_\mathcal{C}} \max_{j=1 \sim N_\mathcal{C}} \left(\frac{\overline{CP_i} + \overline{CP_j}}{\|\mu_{C_i} - \mu_{C_j}\|_2}\right), \quad (6)$$

where $N_\mathcal{C}$ is the number of clusters. Intuitively, the DB index combines the measures of both intra-cluster compactness and inter-cluster separation: a smaller value of $\overline{CP_i}$ indicates a more compact cluster, and a larger value of $\|\mu_{C_i} - \mu_{C_j}\|_2$ indicates a more distinguishable inter-cluster separation between two clusters.

Furthermore, according to the AIC criteria, we choose the model that minimizes

$$AIC = -2\ln(\hat{L}) + 2N_\mathcal{C}. \quad (7)$$

According to the distribution theory of sample statistics [42], the logarithm of maximum likelihood function $(\hat{L})$ represented by the intra-cluster deviation distribution density is defined as

$$\begin{aligned} \ln(\hat{L}) &= \ln\Big(\prod_{i=1}^{N_\mathcal{C}} \frac{|C_i|}{\sum_{i=1}^{N_\mathcal{C}} |C_i|} \Big/ \frac{d_{max} - d_{min}}{N_\mathcal{C}}\Big) \\ &= -N_\mathcal{C} \ln \frac{\sum_{i=1}^{N_\mathcal{C}} |C_i|}{N_\mathcal{C}} - \sum_{i=1}^{N_\mathcal{C}} \ln \frac{d_{max} - d_{min}}{|C_i|}. \end{aligned} \quad (8)$$

where d_{max} and d_{min} denote the maximum and minimum Euclidean distance from node v_i to its assigned cluster centroid μ_{C_i}, respectively.

From the BIC criteria, we know that a higher value of BIC indicates a better model. For consistency, we make it negative to select a model that minimizes

$$BIC = -(\ln(\hat{L}) - \frac{N_\mathcal{C}}{2} \ln(\sum_{i=1}^{N_\mathcal{C}} |C_i|)). \quad (9)$$

We formally define the road trajectory clustering optimization problem as follows:

Definition 9: Given a road network composed of N_n intersections $\{n_1, n_2, \ldots, n_{N_n}\}$, N_s road segments $\{s_1, s_2, \ldots, s_{N_s}\}$, and a set of trajectories $\{T_1, T_2, \ldots, T_{N_\mathcal{T}}\}$, we wish to divide the trajectories into groups $\{C_1, C_2, \ldots, C_{N_\mathcal{C}}\}$ under similarity criteria ($SS(v_i, v_j)$ and $AS(v_i, v_j)$) to achieve $minAIC$, $minBIC$ and $minDB$, i.e.,

$$\min(\overline{CP_i}),$$

and

$$\max(\|\mu_{C_i} - \mu_{C_j}\|_2).$$

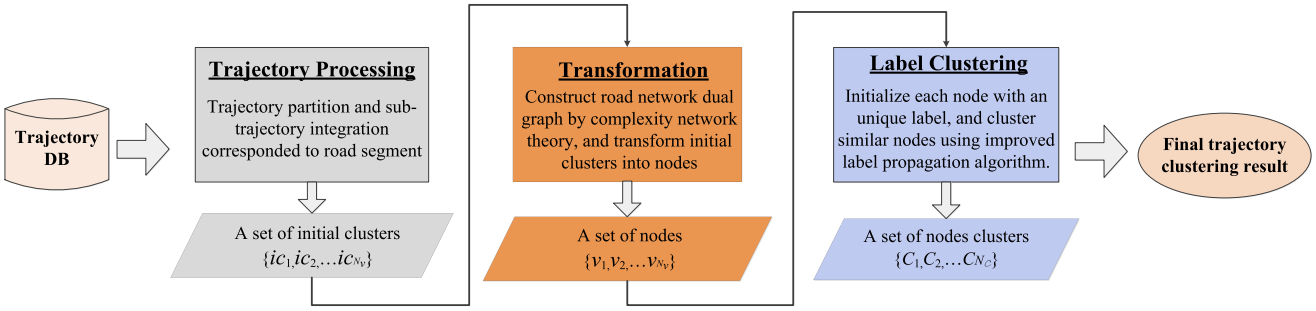


Fig. 6. The framework of LBTC.

In the real world, trajectories could be created by different types of objects, such as animals or hurricanes, which may move in a more free manner. Therefore, distance computation could be the only or the best way to measure the similarity between such trajectories. However, unlike the objects moving freely, vehicle trajectory is characterized by the spatial constraints of road networks, such as road segments and intersections, the traffic flows on consecutive road segments as well as mobile object density. These characters would very likely affect trajectory clustering. For example, without actually observing the flow of traffic, it is unclear whether the trajectories on two road segments should be gathered together. Thus, distance measurement is no longer conducive to clustering vehicle trajectories. Instead, we need a similarity measure that considers a combination of these complex features.

IV. Label-Based Trajectory Clustering

A cluster C_i consists of a group of nodes, each of which represents a initial cluster. As shown in Fig. 6, we design a label-based trajectory clustering approach, referred to as LBTC, consisting of three phases as follows.

In Phase 1 of raw trajectory processing, given a set of moving object trajectories $\{T_1, T_2, \ldots, T_{N_\mathcal{T}}\}$, LBTC divides them into sub-trajectories according to the intersection distribution of the road segment, and then aggregates these sub-trajectories associated with the same road segment, forming a set of initial clusters $\{ic_1, ic_2, \ldots ic_{N_v}\}$

In Phase 2 of transformation, LBTC transforms the road network using the dual graph method, and converts each initial cluster to a node v, which enables us to convert the clustering problem from sub-trajectories in the road network space to nodes in the complex network space.

In Phase 3 of clustering, based on the dual graph topology constructed in Phase 2, we employ the community detection method for clustering to obviate the use of distance measurement. We first design a similarity criteria to improve the accuracy of label propagation clustering, and then perform label-based clustering iteratively in a certain order until all node labels remain unchanged. After that, we gather these nodes associated with the same label together to form clusters $C = \{C_1, C_2, \ldots, C_{N_C}\}$.

Note that Phase 1 and Phase 2 are relatively more straightforward and are briefly discussed in the previous section. Hence, we focus our explanation on Phase 3, which consists of node initialization, label selection, and clustering.

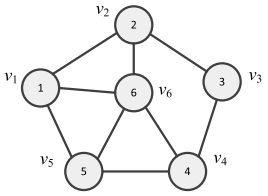


Fig. 7. Initialize node labels.

A. Node Initialization

The Label Propagation Algorithm (LPA), widely used for community structure detection in social networks, is fast, simple and scalable to large networks [43].

In our work, we leverage LPA for trajectory clustering. At initialization, each node is assigned with a unique label before clustering. For the example network in Fig. 7, nodes $\{v_1, v_2, \ldots, v_6\}$ carry six sequential labels $\{1, 2, \ldots, 6\}$. Since the node with a larger density is more likely to be a cluster center, we update the node labels in a descending order of $density$ to reflect the node influences on the cluster.

B. Label Selection

LPA propagates labels using the network structure alone as guide [44] and selects a label with the largest number of neighbors, while ignoring the node attributes. Therefore, some well-defined metrics are needed to capture the essence of similarity between nodes. In most cases, the clustering results would appear more reasonable with the following two considerations: (1) from the perspective of network structure, a node and most of its neighbors should be placed in the same cluster; (2) from the perspective of node attributes, the nodes in the same cluster should have the same or similar attributes.

Based on the above considerations, we extend LPA to support vehicle trajectory clustering by taking into account both the network structure SS (dual graph topology) and node attributes AS (density, flow, and speed), as follows:

$$F(v_i, v_j) = \omega_A \cdot AS(v_i, v_j) + \omega_S \cdot SS(v_i, v_j), \qquad (10)$$

where ω_A and ω_S are the weights of AS and SS, respectively, which satisfy $\omega_A \geq 0$, $\omega_S \geq 0$, and $\omega_A + \omega_S = 1$. The

Algorithm 1 $chooseLabelToUpdate(v_i)$
INPUT: selected node v_i; weight ω_δ; ω_β; ω_A; ω_S
OUTPUT: the *label* of the most similar neighbor

1: $\pi(v_i) \leftarrow getNeighborNodes(v_i)$;
2: calculate $density(v_i)$ and $speed(v_i)$ of v_i, respectively;
3: $speedSum(v_i) \leftarrow$ add up the speed limits of $\pi(v_i)$;
4: **for** each $v_x \in \pi(v_i)$ **do**
5: calculate $flow(v_i, v_x)$ between v_i and v_x;
6: $AS(v_i, v_x) \leftarrow calculateAS(flow(v_i, v_x), speed(v_x), density(v_i), speedSum(v_i))$;
7: $SS(v_i, v_x) \leftarrow calculateSS(\pi(v_i), \pi(v_x))$;
8: calculate $F(v_i, v_x) \leftarrow \omega_A \cdot AS(v_i, v_x) + \omega_S \cdot SS(v_i, v_x)$;
9: select node v_x with $maxF$, i.e., the maximum $F(v_i, v_x)$ value;
10: **return** *label* of node v_x

Algorithm 2 $labelClustering(v)$
INPUT: A set of nodes $\{v_1, v_2, \ldots, v_{N_v}\}$
OUTPUT: A set of clusters $\{C_1, C_2, \ldots, C_{N_C}\}$

1: $isChange = 1$;
2: set a unique label for each node;
3: $orderList \leftarrow setUpdateOrder(\{v_1, v_2, \ldots, v_{N_v}\})$;
4: **while** $isChange == 1$ **do**
5: **for** each $v_i \in orderList$ **do**
6: $count = 0$;
7: record $label\_v_i$ of v_i;
8: $updateLabel \leftarrow chooseUpdateLabel(v_i)$;
9: **if** $updateLabel \neq label\_v_i$ **then**
10: change $label\_v_i$ to $updateLabel$;
11: $count++$;
12: **if** $count = 0$ **then**
13: set $isChange = 0$;
14: place the nodes with the same label in one cluster C_i;
15: $\{C_1, C_2, \ldots, C_{N_C}\} \leftarrow mergerCluster()$;

weights ω_A and ω_S are usually dependent on the application under consideration, and could be set based on an empirical study. In our work, we set these weights to be equal by default, which works well in many applications and datasets.

We design a procedure, $chooseLabelToUpdate(v_i)$, to compute the similarity degree between v_i and its neighbors $\pi(v_i)$. The pseudocode of this procedure is provided in Algorithm 1. It first determines the neighbors $\pi(v_i)$ of v_i, and then computes the values of $AS(v_i, v_x)$ and $SS(v_i, v_x)$ ($v_x \in \pi(v_i)$) with node attributes ($density(v_i)$, $speed(v_i)$ and $flow(v_i, v_x)$) to obtain $F(v_i, v_x)$. Among v_i's neighbours, the algorithm selects the maximum $F(v_i, v_x)$ value denoted as $maxF$, where the label of v_x is the updated label for v_i.

C. Label-Based Clustering

We propose a label-based trajectory clustering procedure, denoted as $labelClustering(v)$, whose pseudocode is provided in Algorithm 2. Given a set of nodes $\{v_1, v_2, \ldots, v_{N_v}\}$ transformed from initial clsuters, $labelClustering(v)$ creates a number of clusters $\{C_1, C_2, \ldots, C_{N_C}\}$. These nodes are first arranged in a descending order of density (line 3), and then propagated iteratively in the $orderList$ sequence until all node labels remain unchanged (lines 5-18). During each iteration, each node determines the most similar neighbor node using $chooseUpdateLabel(v_i)$ (Algorithm 1) and updates its own label to this neighbor's label (lines 6-14). As the labels propagate throughout the network, densely connected groups of nodes reach a consensus on their labels. At the end of this procedure, the nodes carrying an identical label are grouped into the same cluster.

D. Time Complexity Analysis

The proposed LBTC algorithm consists of three main components as shown in Fig. 6. The trajectory processing takes $O(n_p)$ for partitioning and $O(n_{sub})$ for mapping, where n_p is the number of trajectory points in the dataset and n_{sub} is the number of sub-trajectories. In the stage of transformation, there exists a linear scanning of sub-trajectories, which takes $O(n_{sub})$ time. Obviously, the most computationally intensive part of LBTC is the label propagation process. The cost per iteration is dominated by the $chooseUpdateLabel$ function (Algorithm 1) of $O(n_{irs})$ for selecting the most similar neighbor node. Hence, the overall time complexity of LBTC is $O(n_p + 2 \cdot n_{sub} + a \cdot n_{irs})$, or $O(n_{irs})$, where a denotes the number of iterations and n_{irs} denotes the number of the nodes transformed from the initial clusters in LBTC.

TABLE I
NODE ATTRIBUTES

	v_1	v_2	v_3	v_4	v_5	v_6
density	3	3	2	3	2	4
speed	60	65	70	70	65	80
degree	3	3	2	3	3	4

TABLE II
THE FLOW MATRIX OF PAIRWISE NODES

	v_1	v_2	v_3	v_4	v_5	v_6
v_1	0	1	0	0	1	1
v_2		0	1	0	0	1
v_3			0	1	0	0
v_4				0	2	1
v_5					0	0
v_6						0

E. An Illustration of Label Clustering

For illustration purposes, we use a simple example to explain the procedure of the above label-based clustering algorithm. As shown in Fig. 8, the road network dual graph consists of six nodes transformed from Fig. 5. Table I and Table II tabulate the degree and attributes ($density(v_i)$, $speed(v_i)$, $flow(v_i, v_j)$) of each node, respectively, according to the information of Fig. 7. We focus on the iterative procedure followed by node initialization in Section IV-A.

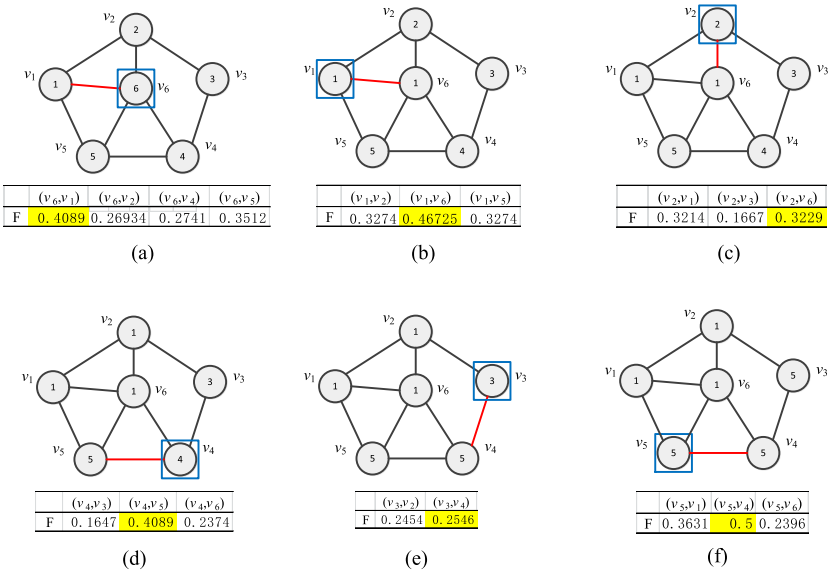


Fig. 8. The procedure of label propagation.

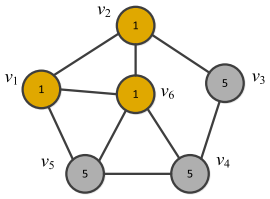


Fig. 9. Final node clusters.

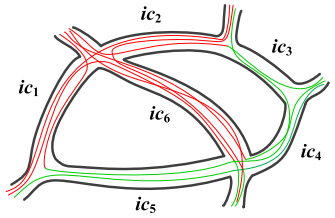


Fig. 10. Final trajectory clusters.

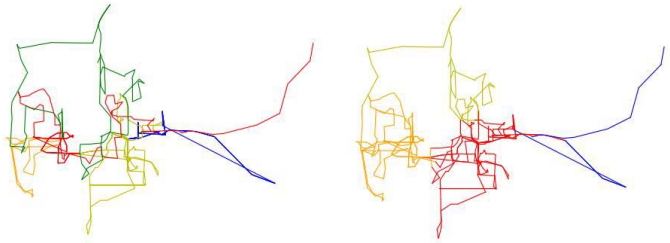
Before the iteration, we first set the update order according to the density values listed in Table I, i.e., $\{v_6, v_1, v_2, v_4, v_3, v_5\}$. Therefore, v_6 is the first node to update its label. The value of the update criteria F, measuring the similarity degree between the selected node (starting from v_6) and its neighbors $\{v_1, v_2, v_4, v_5\}$, is listed under the topological graph, where the $maxF$ value is marked in yellow. Obviously, v_1 is the most similar to v_6 in terms of both structure and attributes. Thus, v_6's label is updated with the label of v_1 (i.e., 1), as shown in Fig. 8 (a). The remaining nodes take turn to update their labels in a similar way, as shown in Fig. 8 (b) – (e), respectively.

After the first iteration, as shown in Fig. 8 (f), we group nodes associated with the same label together to form clusters, Fig. 9 illustrates that the entire network is partitioned into two clusters, marked in different colors.

To facilitate the understanding of label clustering, we transform nodes into initial clusters reversely. As shown in Fig. 10, $\{v_1, v_2, \ldots, v_6\}$ are converted into $\{ic_1, ic_2, \ldots, ic_6\}$, and all trajectories are divided into two clusters $\{ic_1, ic_2, ic_6\}$ and $\{ic_3, ic_4, ic_5\}$ systematically, marked in different colors.

V. Algorithm Implementation and Performance Evaluation

We conduct a simulation-based evaluation of our proposed label-based clustering algorithm in terms of clustering quality and runtime performance in comparison with existing approaches. Specifically, we compare LBTC with NEAT [3] and TRACLUS [20], which are representatives of density-based and distance-based clustering. We provide a brief description for each with focus on their key steps and time complexity analysis.

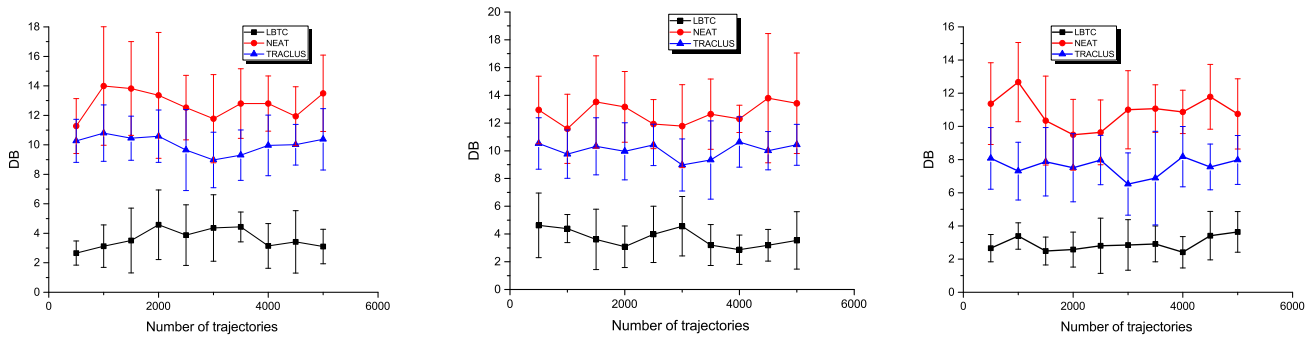


(a) Sample trajectory data (b) Cluster result

Fig. 11. Cluster result for T-drive dataset.

A. Algorithms in Comparison

TRACLUS partitions a trajectory into a series of sub-trajectories and performs DBSCAN [21] to group similar sub-trajectories together. The time cost for partitioning is determined by the number of points in the trajectory dataset. In the DBSCAN phase, the iteration is executed for n_{sub} times, which is the number of sub-trajectories.



(a) DB index for SJ datasets (b) DB index for ATL datasets (c) DB index for T-drive datasets

Fig. 12. Comparison of clustering quality using DB index.

In each iteration, the cost for placing one sub-trajectory in the cluster has two parts: (i) ε-neighborhood query takes $O(\log n_{sub})$ since it employs a spatial index, and (ii) cluster expansion [21] performs a linear scanning for each selected sub-trajectory's neighbors. Therefore, the overall time complexity of TRACLUS is $O(n_p + n_{sub} \cdot (\log n_{sub} + \bar{n}_{\pi_{sub}}))$, i.e., $O(n_{sub} \log n_{sub})$, where $\bar{n}_{\pi_{sub}}$ is the average number of retrieved ε-neighborhoods per sub-trajectory.

NEAT consists of three components, namely, base-cluster, flow-cluster, and opt-cluster [3]. The first stage of base-cluster scans the sequence of points in all the trajectories and integrates the sub-trajectories to form base clusters, which takes $O(n_p + n_{sub})$. The cost for the second stage of flow-cluster is determined by the number of base clusters, while the third stage of opt-cluster is based on DBSCAN and executes ε-neighborhood queries as the cluster expansion in TRACLUS. Hence, the time complexity of NEAT is $O(n_p + n_{sub} + n_{irs} + n_{flow} \cdot (\log n_{flow} + \bar{n}_{\pi_{flow}}))$, i.e., $O(n_{flow} \log n_{flow})$, where n_{irs} denotes the number of the base clusters corresponding to road segments in NEAT, n_{flow} denotes the number of flow-clusters [3] ($n_{flow} < n_{sub}$), and $\bar{n}_{\pi_{flow}}$ denotes the average number of retrieved ε-neighborhoods per flow cluster.

B. Experimental Settings

We use two trajectory datasets for performance evaluation: real-world T-Drive dataset [45] and synthetic dataset.

1) Real-World T-Drive Dataset: This dataset contains the GPS trajectories of 10,357 taxis during the period of Feb. 2 to Feb. 8, 2008 in Beijing, China. There are about 15 million points in this dataset and the total distance of the trajectories spans 9 million kilometers. In our experiments, we generate different test datasets, ranging from 500 to 5,000 trajectories randomly selected from the T-Drive dataset.

2) Synthetic Dataset: For accuracy test, we generate synthetic datasets using the event-based simulator GTMobiSim [46], which randomly generates moving object traces on real-life road maps of West San Jose (SJ) and North West Atlanta (ATL) [47]. The ATL map is a rural/suburban road network with typical low density characteristics. The SJ map is a suburban road network with a denser network topology, with a higher density of road segments and junctions [48]. In our experiments, we generate different test datasets, ranging from 500 to 5,000 trajectories for each road map.

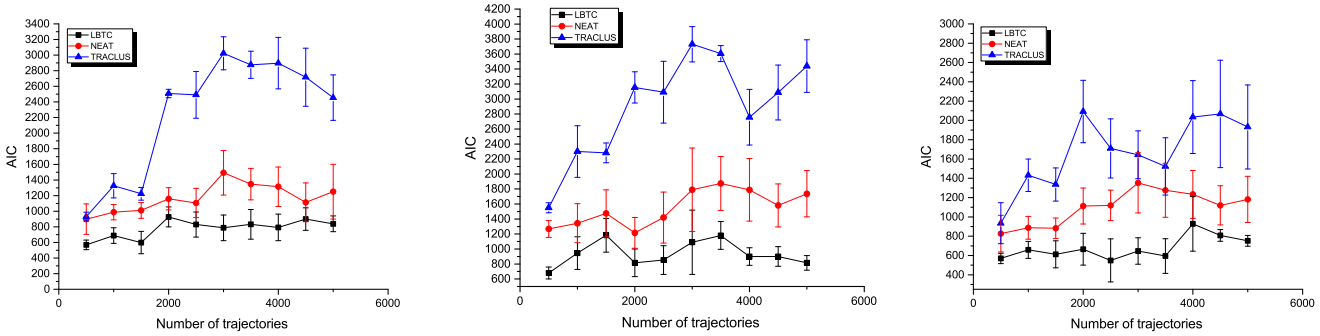
All algorithms are implemented in Java 1.7, and all experiments are conducted on a Windows PC workstation equipped with Intel(R) Core(TM) i5-4590 CPU@3.30GHz and 8GB of memory.

C. Visualization of LBTC Clustering Results

Considering that the original T-Drive dataset is enormously large, dense, and complicated, we run LBTC on 5 taxi ID (totalling 3,890 points) randomly selected from the dataset. We provide the results in Fig. 11, where Fig. 11(a) plots the mobility traces in different colors, and Fig. 11(b) shows that LBTC discovers four spatial clusters as groups of sub-trajectories, which represent both dense and highly continuous flows of mobile objects. These results show that the red and orange clusters are discovered in the two most dense regions as expected. Moreover, we observe that the blue cluster is not dense enough to be identified as a cluster. Thus, if we perform filtering on the obtained clusters by thresholding the number of sub-trajectories, the remaining clusters would well represent the regions or road segments with a high volume of traffic to help optimize route planning, transit stop selection, and so on.

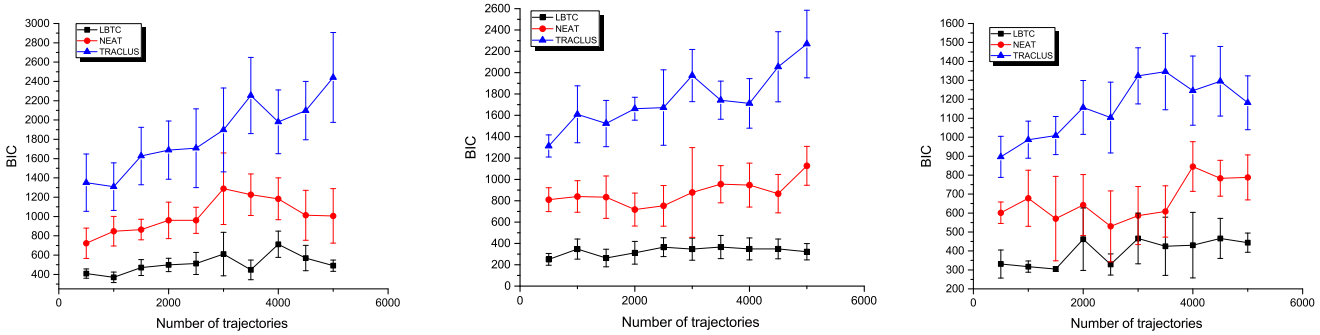
D. Clustering Quality Evaluation

We evaluate the clustering quality of LBTC in comparison with TRACLUS and NEAT in terms of Davies-Bouldin Index (DB), Akaike Information Criterion (AIC), and Bayesian Information Criterion (BIC). Note that DB, AIC, and BIC are inversely proportional to the clustering quality, as introduced in Section III-C, i.e., a smaller value indicates a better clustering quality. We run these three algorithms with different numbers of trajectories ranging from 500 to 5000 at an interval of 500. For each number of trajectories, we generate 10 random trajectory datasets. Fig. 12, Fig. 13 and Fig. 14 plot the mean and standard deviation of the DB, AIC and BIC indices across 10 different datasets for each number of trajectories, respectively. We observe that LBTC achieves a better clustering quality than NEAT and TRACLUS for both T-Drive and synthetic datasets. Note that both NEAT and TRACLUS rely on the DBSCAN [21] algorithm, making it very challenging



(a) AIC for SJ datasets (b) AIC for ATL datasets (c) AIC for T-drive datasets

Fig. 13. Comparison of clustering quality using AIC.



(a) BIC for SJ datasets (b) BIC for ATL datasets (c) BIC for T-drive datasets

Fig. 14. Comparison of clustering quality using BIC.

to set distance parameters EPS and $MinPts$, especially when applied to the dataset whose density varies widely. In general, time-varying vehicle trajectories often result in uneven data distributions, which render the traditional density-based clustering approaches such as NEAT and TRACLUS unsuitable for trajectory clustering in real-life road networks. However, LBTC without distance parameter is more suitable to process uneven data distributions.

We further compare the clustering quality of LBTC with the optimal solutions on small-scale SJ datasets ranging from 5 to 15 trajectories. For each size, we randomly generate 10 instances with different node attributes (density, flow, speed) and dual graph topologies. For each instance, we calculate the corresponding DB index in this range. We then run all of the three algorithms on these instances and compare the DB results with OPT-LBTC, which uses an exhaustive search approach to produce the optimal result (minDB) of all possible clustering structures. Fig. 15 plots the mean and standard deviation of DB indices across 10 instances for each number of trajectories. We observe that LBTC achieves close-to-optimal results in these small-scale problem instances. These simulation results demonstrate that LBTC is able to produce a better clustering structure than TRACLUS and NEAT for both small-scale datasets and large-scale datasets.

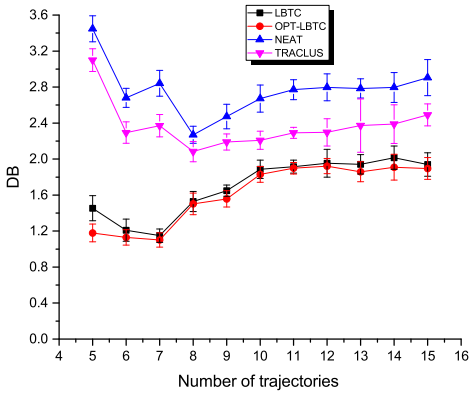
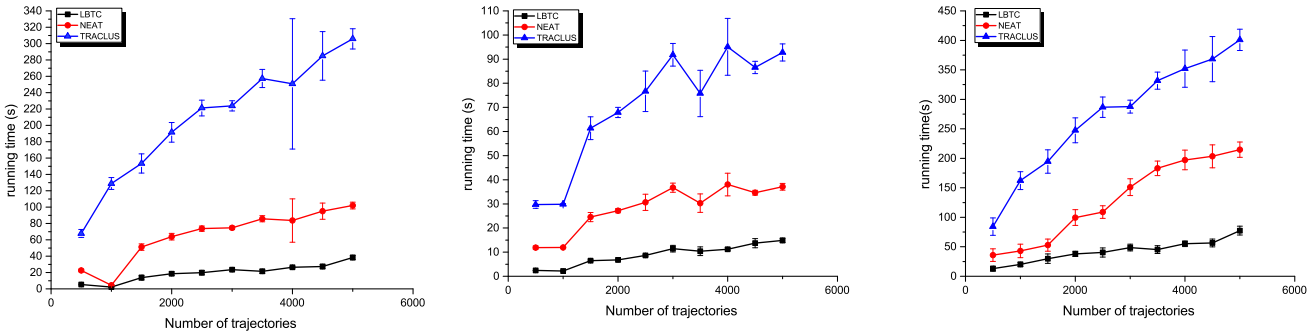


Fig. 15. Comparison with the optimum DB index on small-scale SJ datasets.

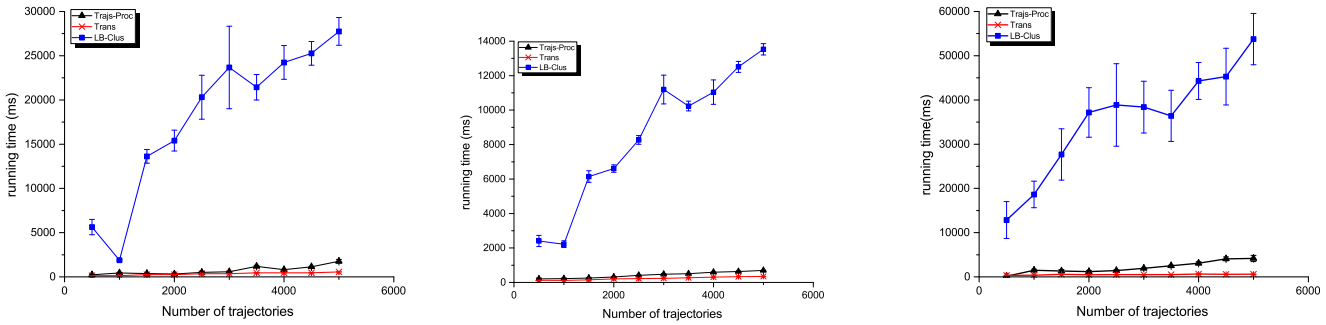
E. Running Time Performance Evaluation

We also compare the proposed clustering algorithm with TRACLUS and NEAT in terms of running time. We run these three algorithms with different numbers of trajectories on the synthetic SJ, ATL dataset and the real-world T-drive dataset, ranging from 500 to 5000 at an interval of 500 trajectories. Again, for each number of trajectories, we generate 10 random trajectory datasets. We plot the mean and standard deviation of the algorithm running time across 10 datasets for each number of trajectories in Fig. 16. These results show that LBTC runs significantly faster than the other two algorithms in comparison for both SJ, ATL datasets and T-drive datasets, which is consistent with our time complexity analysis, and the superiority of LBTC becomes more obvious as the number of trajectories increases. Also, we observe that, with the benefit



(a) SJ datasets (b) ATL datasets (c) T-drive datasets

Fig. 16. Comparison of algorithm running time.



(a) SJ datasets (b) ATL datasets (c) T-drive datasets

Fig. 17. Running time of the three phases of LBTC.

of high efficiency of LPA, LBTC is over six times faster than TRACLUS and approximately three times faster than NEAT.

The experimental results in terms of both clustering quality and execution efficiency as shown in Fig. 12 - Fig. 16 indicate that the proposed LBTC algorithm takes significantly less time to produce a better clustering structure than the traditional TRACLUS and NEAT approaches.

We further analyze the time spent on the execution of the three phases of LBTC for both SJ, ATL and T-drive datasets, ranging from 500 to 5000 trajectories. We plot the mean and standard deviation of the phase running time across 10 different datasets for each number of trajectories in Fig. 17, where the three phases of Trajectory Processing, Transformation, and Label-Based Clustering are labeled as "Trajs-Pro", "Trans", and "LB-Clus", respectively. Phase 1 of "Trajs-Pro" is a simple and fast procedure to partition and organize trajectories to initial clusters corresponding to road segments. Phase 2 of "Trans" transforms initial cluster to nodes and constructs a dual graph topology, which incurs a similar amount of time as Phase 1. Phase 3 of "LB-Clus" is the kernel of our clustering algorithm, and takes longer than the first two phases. However, the execution time of "LB-Clus" is almost linear with respect to the number of trajectories, indicating that LBTC scales well to larger datasets.

VI. Conclusion

We proposed a novel label-based trajectory clustering algorithm, refer to as LBTC, which leverages the complex network theory to obviate the need to compute trajectory distance. Extensive experiments demonstrate that LBTC significantly improves both clustering quality and runtime performance over the existing methods. The proposed trajectory labeling approach has wide applications in various traffic and location service systems, including traffic navigation, urban planning, service recommendation, military scheduling, traffic control, logistics and distribution, and vehicle monitoring.

References

[1] E. Tiakas, A. N. Papadopoulos, A. Nanopoulos, Y. Manolopoulos, D. Stojanovic, and S. Djordjevic-Kajan, "Searching for similar trajectories in spatial networks," *J. Syst. Softw.*, vol. 82, no. 5, pp. 772–788, 2009.

[2] M. K. El Mahrsi and F. Rossi, "Graph-based approaches to clustering network-constrained trajectory data," in *New Frontiers in Mining Complex Patterns*, vol. 7765. Berlin, Germany: Springer, 2013, pp. 124–137.

[3] B. Han, L. Liu, and E. Omiecinski, "NEAT: Road network aware trajectory clustering," in *Proc. IEEE Int. Conf. Distrib. Comput. Syst.*, Jun. 2012, pp. 142–151.

[4] J.-R. Hwang, H.-Y. Kang, and K.-J. Li, "Spatio-temporal similarity analysis between trajectories on road networks," in *Proc. Int. Conf. Perspect. Conceptual Modeling*, 2005, pp. 280–289.

[5] P. C. Besse, B. Guillouet, J.-M. Loubes, and F. Royer, "Review and perspective for distance-based clustering of vehicle trajectories," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 11, pp. 3306–3317, Nov. 2016.

[6] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series," in *Proc. Int. Conf. Knowl. Discovery Data Mining*, 1994, pp. 359–370.

[7] M. Vlachos, G. Kollios, and D. Gunopulos, "Discovering similar multidimensional trajectories," in *Proc. Int. Conf. Data Eng.*, 2002, p. 673.

[8] L. Chen and R. Ng, "On the marriage of Lp-norms and edit distance," in *Proc. 13th Int. Conf. Very Large Data Bases*, 2004, pp. 792–803.

[9] L. Chen, V. Oria, and M. T. Öria, "Robust and fast similarity search for moving object trajectories," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2005, pp. 491–502.

[10] F. Hausdorff, *Grundzüge der Mengenlehre*. Amer. Math. Soc., 1978.

[11] M. M. Fréchet, "Sur quelques points du calcul fonctionnel," *Rendiconti Circolo Matematico Palermo*, vol. 22, no. 1, pp. 1–72, 1906.

[12] B. Lin and J. Su, "Shapes based trajectory queries for moving objects," in *Proc. ACM Int. Workshop Geograph. Inf. Syst.*, 2005, pp. 21–30.

[13] M. Barthélemy, "Spatial networks," *Phys. Rep.*, vol. 499, nos. 1–3, pp. 1–101, 2011.

[14] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 76, no. 3, 2007, Art. no. 036106.

[15] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-1, no. 2, pp. 224–227, Apr. 1979.

[16] H. Akaike, "A new look at the statistical model identification," *IEEE Trans. Autom. Control*, vol. AC-19, no. 6, pp. 716–723, Dec. 1974.

[17] J. Anděl, M. G. Perez, and A. I. Negrao, "Estimating the dimension of a linear model," *Kybernetika*, vol. 17, no. 6, pp. 514–525, 1981.

[18] M. Nanni and D. Pedreschi, "Time-focused clustering of trajectories of moving objects," *J. Intell. Inf. Syst.*, vol. 27, no. 3, pp. 267–289, 2006.

[19] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "OPTICS: Ordering points to identify the clustering structure," in *Proc. Int. Conf. Manage. Data*, 1999, pp. 1–12.

[20] J. G. Lee, J. Han, and K. Y. Whang, "Trajectory clustering: A partition-and-group framework," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2007, pp. 593–604.

[21] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. Kdd*, 1996, vol. 96, no. 34, pp. 226–231.

[22] B. Han, L. Liu, and E. Omiecinski, "Road-network aware trajectory clustering: Integrating locality, flow, and density," *IEEE Trans. Mobile Comput.*, vol. 14, no. 2, pp. 416–429, Feb. 2015.

[23] N. B. Ghrab, E. Fendri, and M. Hammami, "Clustering-based abnormal event detection: Experimental comparison for similarity measures' efficiency," in *Proc. Int. Conf. Image Anal. Recognit.*, 2016, pp. 367–374.

[24] I. Sanchez, Z. M. M. Aye, B. I. P. Rubinstein, and K. Ramamohanarao, "Fast trajectory clustering using hashing methods," in *Proc. Int. Joint Conf. Neural Netw.*, 2016, pp. 3689–3696.

[25] S. Ingrassia, S. C. Minotti, and G. Vittadini, "Local statistical modeling via a cluster-weighted approach with elliptical distributions," *J. Classification*, vol. 29, no. 3, pp. 363–401, 2012.

[26] J. D. Banfield and A. E. Raftery, "Model-based Gaussian and non-Gaussian clustering," *Biometrics*, vol. 49, no. 3, pp. 803–821, 1993.

[27] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.

[28] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Hierarchical Dirichlet processes," *J. Amer. Statist. Assoc.*, vol. 101, no. 476, pp. 1566–1581, Dec. 2006.

[29] Z. Chen, Y. Yan, and T. Ellis, "Lane detection by trajectory clustering in urban environments," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, Oct. 2014, pp. 3076–3081.

[30] K. Blekas, N. Galatsanos, and A. Likas, "A sparse regression mixture model for clustering time-series," in *Proc. Hellenic Conf. Artif. Intell.*, 2008, pp. 64–72.

[31] X. Wang, X. Ma, and W. E. Grimson, "Unsupervised activity perception in crowded and complicated scenes using hierarchical Bayesian models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 3, pp. 539–555, Mar. 2009.

[32] X. Wang, K. T. Ma, G. W. Ng, and W. E. L. Grimson, "Trajectory analysis and semantic region modeling using a nonparametric Bayesian model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2008, pp. 1–8.

[33] X. Wang, K. T. Ma, G.-W. Ng, and W. E. L. Grimson, "Trajectory analysis and semantic region modeling using nonparametric hierarchical Bayesian models," *Int. J. Comput. Vis.*, vol. 95, no. 3, pp. 287–312, 2011.

[34] H. Wang, J. Ondrej, and C. O'Sullivan, "Path patterns: analyzing and comparing real and simulated crowds," in *Proc. ACM SIGGRAPH Symp. Interact. 3D Graph. Games*, 2016, pp. 49–57.

[35] A. Rodriguez and A. Laio, "Machine learning. Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, p. 1492, 2014.

[36] D. Easley and J. Kleinberg, *Networks, Crowds, and Markets: Reasoning about a Highly Connected World*. Cambridge, U.K.: Cambridge Univ. Press, 2010.

[37] M. Weber, L. Liu, K. Jones, M. J. Covington, L. Nachman, and P. Pesti, "On map matching of wireless positioning data: A selective look-ahead approach," in *Proc. ACM SIGSPATIAL Int. Symp. Adv. Geograph. Inf. Syst. (ACM-GIS)*, San Jose, CA, USA, Nov. 2010, pp. 290–299.

[38] J. Crnic, *Introduction to Modern Information Retrieval*. New York, NY, USA: McGraw-Hill, 1983.

[39] N. R. Pal and J. Biswas, "Cluster validation using graph theoretic concepts," *Pattern Recognit.*, vol. 30, no. 6, pp. 847–857, 1997.

[40] G. Gan, C. Ma, and J. Wu, *Data Clustering: Theory, Algorithms, and Applications*. Philadelphia, PA, USA: SIAM, 2007.

[41] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, Sep. 1999.

[42] S. Venkatraman, J. Caffery, Jr., and H.-R. You, "Location using LOS range estimation in NLOS environments," in *Proc. IEEE Veh. Technol. Conf.*, May 2002, pp. 856–860.

[43] S. Mostafavi, A. Goldenberg, and Q. Morris, "Labeling nodes using three degrees of propagation," *PLoS ONE*, vol. 7, no. 12, 2012, Art. no. e51947.

[44] R. Hosseini and R. Azmi, "Memory-based label propagation algorithm for community detection in social networks," in *Proc. Int. Symp. Artif. Intell. Signal Process.*, 2015, pp. 256–260.

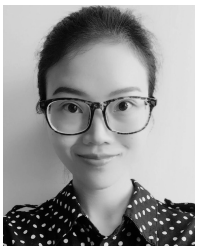
[45] Y. Zheng. (Aug. 2011). *T-Drive Trajectory Data Sample*. [Online]. Available: https://www.microsoft.com/en-us/research/publication/t-drive-trajectory-data-sample/

[46] *Gtmobisim*. Accessed: Oct. 13, 2017. [Online]. Available: http://code.google.com/p/gt-mobisim/

[47] *USGS Data*. Accessed: Feb. 25, 2017. [Online]. Available: http://edc2.usgs.gov/geodata/ index.php

[48] B. Han, L. Liu, and E. Omiecinski, "A systematic approach to clustering whole trajectories of mobile objects in road networks," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 5, pp. 936–949, May 2017.

Xinzheng Niu received the Ph.D. degree in computer science from the University of Electronic Science and Technology of China, Chengdu, Sichuan, China. He is currently an Associate Professor with the School of Computer Science and Engineering, University of Electronic Science and Technology of China. His main research interests include data mining and information security. He is a member of CCF.



Ting Chen is currently pursuing the M.S. degree with the School of Computer Science and Engineering, University of Electronic Science and Technology of China. Her research interests include social networks and data mining.

Chase Q. Wu received the Ph.D. degree in computer science from Louisiana State University in 2003. He was a Research Fellow with the Oak Ridge National Laboratory from 2003 to 2006 and an Assistant and Associate Professor with the University of Memphis from 2006 to 2015. He is currently an Associate Professor with the Department of Computer Science, New Jersey Institute of Technology. His research interests include big data, distributed and parallel computing, and computer networks.

Jiajun Niu is currently pursuing the M.S. degree with the School of Computer Science and Engineering, University of Electronic Science and Technology of China. Her research interests include big data and data mining.

Yuran Li is currently pursuing the M.S. degree with the School of Information and Software Engineering, University of Electronic Science and Technology of China. Her research interests include big data and data mining.