Making Sense of Trajectory Data: A Partition-and-Summarization Approach

Han Su[†] Kai Zheng[#] Kai Zeng[‡] Jiamin Huang[+] Shazia Sadiq[†] Nicholas Jing Yuan[*] Xiaofang Zhou[#]

[†#] *University of Queensland, Australia* [‡]*University of Canifornia, Los Angeles, United States*
[+]*University of Michigan, Ann Arbor, United States* [*]*Microsoft Research Asia, China*
[†]h.su1@uq.edu.au [#]{kevinz, shazia, zxf}@itee.uq.edu.au [‡]kzeng@cs.ucla.edu
[+]jiamin@umich.edu [*]nicholas.yuan@microsoft.com

***Abstract*—Due to the prevalence of GPS-enabled devices and wireless communication technology, spatial trajectories that describe the movement history of moving objects are being generated and accumulated at an unprecedented pace. However, a raw trajectory in the form of sequence of timestamped locations does not make much sense for humans without semantic representation. In this work we aim to facilitate human's understanding of a raw trajectory by automatically generating a short text to describe it. By formulating this task as the problem of adaptive trajectory segmentation and feature selection, we propose a partition-and-summarization framework. In the partition phase, we first define a set of features for each trajectory segment and then derive an optimal partition with the aim to make the segments within each partition as homogeneous as possible in terms of their features. In the summarization phase, for each partition we select the most interesting features by comparing against the common behaviours of historical trajectories on the same route and generate short text description for these features. For empirical study, we apply our solution to a real trajectory dataset and have found that the generated text can effectively reflect the important parts in a trajectory.**

I. Introduction

Driven by major advances in sensor technology, GPS-enabled mobile devices and wireless communications, a large amount of data recording the motion history of moving objects, known as *trajectories*, are currently being generated and managed in scores of application domains. In the past few years, a lot of research works focused on the trajectory analyzing. Effective index structures [35], [26], [3], [25], [5], [10], [34], [6], [14], [27] are built to manage trajectories and support high performance trajectory queries. Data mining methods are applied on trajectories to detect important points of interest (POI) and find the popular route from a source to a destination [17], [16], [15], [19], [40]. Attentions are also drawn to semantic representation or interpretation of trajectory data by associating or annotating GPS locations with semantic entities [38], [30]. Despite the huge efforts which have been made on trajectory management and mining, trajectory data itself is still hard for humans to understand. Table I demonstrates how a raw trajectory is represented in databases, which is a sequence of triples ⟨longitude, latitude, timestamp⟩. In order to facilitate better interpretation of raw trajectories, researchers have proposed several models by associating GPS locations with semantic entities such as POIs, roads, regions, resulting in semantic trajectories or annotated trajectories [38], [30]. Fig. 1(a) demonstrates how the raw trajectory in Table I is shown on a map. From Fig.1(a) we can see that graphical interpretation of the trajectories makes it easier for humans to understand the travel history of this moving object. Nevertheless semantic trajectories have their disadvantages in terms of expressiveness and data volume.

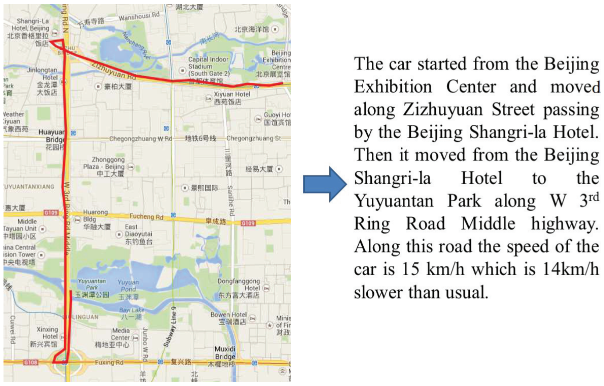
TABLE I. Trajectory in database

Latitude	Longitude	Time-stamp
39.9383	116.339	20131102 09:17:56
39.9382	116.337	20131102 09:18:02
· · ·	· · ·	· · ·
· · ·	· · ·	· · ·
39.9259	116.310	20131102 09:33:26
39.9253	116.310	20131102 09:34:31

- **Expressivity.** Semantic trajectories cannot intuitively express the travel behaviours relating to temporal attributes such as overspeed, sharp speed change, long stopover, etc. Moreover, they cannot highlight the 'interesting' parts of the trajectories such as significant landmarks and important roads. Although all these information have been encoded in semantic trajectories already, it needs substantial manual efforts and expertise to find out.
- **Data volume.** Essentially a semantic trajectory is an enriched version of the raw trajectory, i.e., each space-time point is attached with a set of semantic attributes. Therefore the volume of semantic trajectories can be excessive for storage, processing and communication.

To address these drawbacks of semantic trajectories, we take the philosophy from text summarization in the field of information retrieval, and propose a partition-and-summarization framework. More specifically, given a raw trajectory and external semantic information as inputs, our framework will automatically generate a short text to highlight the significant semantic behaviour of this trajectory. Fig. 1(b) exemplifies the expected summarization for the given trajectory. We find there are several benefits by translating a raw trajectory into text. First, the information conveyed in the text are strategically focused on the most 'interesting' parts of the trajectories, thus making more sense for humans. Second, as the output of our framework is a summarization rather than transformation of raw trajectories (like semantic trajectories), the output text is lightweight and easy to store and communicate. Third, trajectories collected from different sources may have different formats and schema, but they can all be translated to texts

978-1-4799-7964-6/15/$31.00 © 2015 IEEE



The car started from the Beijing Exhibition Center and moved along Zizhuyuan Street passing by the Beijing Shangri-la Hotel. Then it moved from the Beijing Shangri-la Hotel to the Yuyuantan Park along W 3rd Ring Road Middle highway. Along this road the speed of the car is 15 km/h which is 14km/h slower than usual.

(a) Trajectory aligned on a map (b) Summary of the trajectory

Fig. 1. Making sense of trajectory data

with similar style. Fourth, various well-studied text mining algorithms can be applied to the trajectory summaries and valuable semantic knowledge may be discovered.

Trajectory summarization can be used in many applications. By embedding the trajectory summarization technique in GPS modules of cars and cells, an infraction reminder can be created. Every time some driving infractions occur, the driver can receive the infraction travel summary. Also, during traveling, an automatically generated trajectory summary is a good travel dairy, which can be shared to friends via Twitter or Facebook. In the meantime, the trajectory summary can assist blind people to understand the travel history using TTS (Text To Speech) techniques.

Nevertheless this task is faced with several challenges. A whole trajectory seldom has the same travel behaviors uniformly. Usually a trajectory consists of several parts where each part has very different travel behaviors. For example, in Fig. 1(b) the vehicle travels in normal speed in the first half of the trajectory, but in significantly lower speed than normal in the second half. Hence the trajectory should be partitioned into two parts and described separately. Thus, how to properly partition the trajectory is the first question. However, even within a trajectory partition, there are many kinds of travel behaviors to be described, It is non-trivial how to choose the most interesting or significant travel behaviors of each partition. To tackle these challenges, we propose a partition-and-summarization framework: (1) The partition phase tries to find an optimal partition by minimizing the variation of predefined features for the trajectory segments within the same partition. Through this partition, we are able to use more compact representation to summarize each partition. (2) The summarization phase exploits the common patterns learned from historical trajectories, to measure the unusualness of each feature, and generate textual description for the most unusual features with a predefined template. We also implemented this framework in a prototype system—STMaker [32].

To sum up, we make the following major contributions in this paper.

- We identify the limitations of raw and semantic trajectories in terms of their interpretability by humans, and take the initiative to describe individual trajectory with summary texts automatically.
- We develop a partition-and-summarize framework to tackle several challenges of our proposals including granularity control and feature selection.
- We conduct extensive experiments based on a large-scale real trajectory dataset, which empirically demonstrates that the generated textual descriptions can reflect the most significant features of trajectories and are easier for humans to understand.

The remainder of this paper is organized as follows. Section II introduces the preliminary concepts and the trajectory preprocessing. Major features used in the paper are presented in III. We elaborate the details of the two phases, trajectory partitioning and feature selection,in Section IV and Sec. V respectively. We discuss how to construct summary text from features and how to extend customized features to the system in Section VI. The experimental observations are presented in Section VII, followed by a brief review of related work in Section VIII. Section IX concludes the paper and outlines some future work.

II. Problem Statement

In this section, we introduce some preliminary concepts, and formally define the summarization process. Table II summarizes the major notations used in the rest of the paper.

TABLE II. Summarize of notations

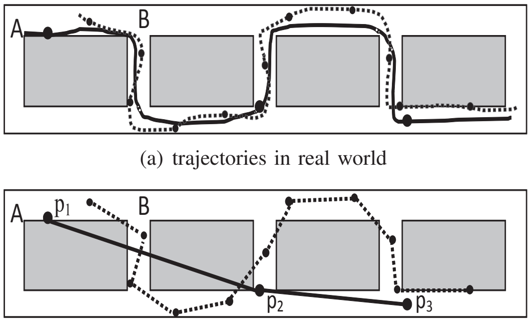
Notation	Definition
T	a raw trajectory
l	a landmark in the space
$l.s$	the significance of a landmark l
$\overline{T}$	a symbolic trajectory
$\overline{TS}_i$	the trajectory segment connecting two consecutive landmarks l_i and l_{i+1} of $\overline{T}$
$\overline{TP}$	a trajectory partition of trajectory $\overline{T}$
f	a feature of trajectory
$\mathbb{F}_{\overline{T}}$	the concerning features of a trajectory $\overline{T}$
$f(\overline{TS})$	$\overline{TS}$'s value of feature f

A. Preliminary Concepts

Definition 1 (Raw Trajectory): A trajectory T is a finite sequence of locations sampled from the original route of a moving object and their associated time-stamps, i.e., $T = [(p_1, t_1), (p_2, t_2), \cdots, (p_n, t_n)]$.

A raw trajectory is represented as a discrete sequence of locations sampled from the continuous movement of the moving object. However, since the sampling strategies used to generate the trajectory data can vary significantly in different sampling methods and sampling rates (e.g., locations could be sampled every 100 meters, or every 5 seconds), the raw trajectory data are not directly usable for summarization purpose, due to the following two reasons: (1) Intuitively, despite of different sampling strategies, different trajectories sampled from the same route should result in the same or similar summarization. However, trajectories generated from the same route could be very different. For example, Fig. 2(a) demonstrates two moving objects following highly similar routes in an urban area, but adopting different sampling strategies. Fig. 2(b) illustrates the actual trajectory data of these two routes stored in the database.

It is easy to observe that the two trajectories T_A and T_B are much more different than they are supposed to be. Such limitation has already been observed by a previous work [31]. Actually, the same route could result in very different raw trajectories under different sampling strategies, which thus leads to undesirably different summaries and are very hard for human to recognize. (2) The locations in trajectories are usually described by latitudes and longitudes. However, these physical positions can hardly give people any intuitive view about the actual route of the moving object, and thus cannot serve as an description in the summary.



(a) trajectories in real world

(b) trajectories stored in database

Fig. 2. Example of trajectory in real world and in database

Therefore, in this paper, we propose to preprocess the raw trajectories using semantic locations which are independent of any raw trajectories. The following summarization is applied on these transformed raw trajectories, and thus can give correct summarization of the original route of the moving object, without being affected by the chosen sampling strategy. These semantic locations are termed as *landmark*.

Definition 2 (Landmark): A landmark l is a geographical point in the space, which is stable and independent of trajectories.

A landmark can be either a Point Of Interest (POI) or a turning point of the road network. We employ our previous research results on anchor-based trajectory calibration [31] to rewrite the raw trajectory T into a landmark-based trajectory $\overline{T}$, by treating landmarks as anchor points. The trajectories so obtained are called *Symbolic Trajectories*. The size $|\overline{T}|$ denotes the number of landmarks of $\overline{T}$.

Definition 3 (Symbolic Trajectory): A symbolic trajectory $\overline{T}$ is a sequence of landmarks and their corresponding timestamps, i.e., $\overline{T} = [(l_1, t_1),\ (l_2,\ t_2),\ \cdots, (l_m, t_m)]$.

Definition 4 (Trajectory Segment): An segment $\overline{TS}_i$ of a symbolic trajectory $\overline{T}$ is a sub-trajectory which connects two consecutive landmarks l_i and l_{i+1} of $\overline{T}$.

For a given symbolic trajectory $\overline{T} = [l_1, l_2, \cdots, l_n]$, $\overline{T}$ has $|\overline{T}| - 1$ segments $\overline{TS}_1$, $\overline{TS}_2$, $\cdots$, $\overline{TS}_{n-1}$. These segments are the basic atoms constructing $\overline{T}$. Two segments are named *contiguous segments* if they share the same landmark as the start and the destination respectively, i.e., $\overline{TS}_2$ and $\overline{TS}_3$ sharing landmark l_3.

In the rest of the paper, the proposed summarization method is applied to symbolic trajectories. Thus, we will use trajectory and symbolic trajectory interchangeably whenever the context is clear.

B. System Overview

Recall the example in Fig. 1(b). In real-life scenarios, people usually describe their trips in the following manner: first divide the whole route into several partitions with significant starting (sources) and ending places (destinations); and then use some significant events to describe the unusual behaviours within each partition. For example, when summarizing the route in Fig. 1(b), Beijing Shangri-la Hotel and Yuyuantan Park are two significant POIs, which mark out a trajectory partition of the route; along the partition, the car was driving on a high way, and was moving 14km/h slower than usual, which are some significant behaviors describing how the car traveled. STMaker follows exactly the same way of how humans think that it automatically generate a summary text for a given trajectory in 4 steps, as shown in Fig. II-B:

1) STMaker rewrites the raw trajectory into a symbolic trajectory.
2) STMaker conducts a partitioning to split the trajectory into several non-overlapping parts. During partitioning, we take consideration of multiple features describing the trajectory. The optimization goal of this phase is to (1) maximize the significance of the landmarks at the two ends of each partition, (2) maximize the information cohesion of each partition in routing and moving features and (3) minimize the information variance in routing and moving features within the same partition. We will detail the partitioning process in Sec. IV.
3) Given the fact that there are too many features to describe, STMaker will choose the most significant features within each partition according to a novel measurement of the interestingness for each feature. E.g., in Fig. 1(b), 'driving on a high way' and 'moving 14km/h slower than usual' are a routing feature and a moving feature, respectively. It will be detailed in Sec. V.
4) The selected features will be plugged into the predefined phrase templates to form the summary for each trajectory partition.

We expect that the summary of trajectory T could give people an intuitive view about the moving behaviour embedded in T, which is critical to endow the dreary and elusive GPS data with comprehensible description. Ideally, given any two trajectories, the differences in their moving behaviours should be reflected in their summary.

III. Feature Extraction

In this section, we present the main features that will be used to describe the trajectories. The features considered in STMaker can be mainly divided into two types: routing features (which describe where the moving object travels), and moving features (which describe how the moving object travels).

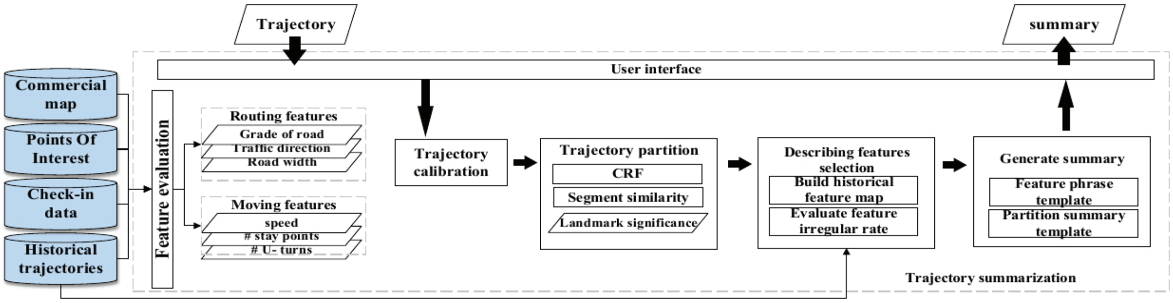


Fig. 3. Framework overview

TABLE III. ROUTING FEATURES

Feature Type	Example	Numeric
grade of road	1 (national highway)	No
road width	13 metres	Yes
traffic direction	2 (one-way road)	No

TABLE IV. MOVING FEATURES

Feature Type	Example	Numeric
speed	86.2 km/h	Yes
# stay points	0	Yes
# U-turns	1	Yes

A. Routing Features

Routing features describe the characteristics related to where the moving object traveled. Thus, as we focus on trajectories collected from vehicles, the natural routing feature is the information about the roads they travel on. For example, with road information, we can tell whether a vehicle moves on highway or not. More importantly, road information can directly affect the moving patterns of the trajectories, for example, people tend to move faster on a highway than on a local road. In the STMaker system, we identify and use 3 kinds of road information (*'grade of road'*, *'road width'* and *'direction'*) as the routing features, shown in the Table III. These features can be extracted from the digital map we have, and well distinguish different kinds of roads. Notably, the value of road width feature is numeric, while the values of grade of road feature and traffic direction feature are categorical. We assign different integers for the categorical features.

Grade of Road: The grade of road is the road type of a trajectory segment $\overline{TS}$. There are seven grades of road: 1 (highway), 2 (express road), 3 (national road), 4 (provincial road), 5 (country road), 6 (village road) and 7 (feeder road). The roads with higher grade (smaller numerical value) usually have higher transportation capacity.

Road Width: The road width of a trajectory segment $\overline{TS}$ is a important feature which can affect how popular the road is, and how fast people can travel on the road.

Direction: The direction indicates the traffic direction of the road. There are two values of direction, i.e., 1 (two-way road) and 2 (one-way road). If an object moves along a one-way road, then one of the most distinctive information of the trajectory is 'moving along a one-way road', while most trajectories move along two-way roads.

B. Moving Features

Moving features indicate how the moving object travels. Many works [4], [41] have been devoted to extracting various moving information from trajectories. Notably, the algorithms extracting moving features need to be applied on the sample-based trajectory instead of the symbolic trajectory. In our system, we propose three types of moving features (*'speed'*, *'number of stay points'*, and *'number of U-turns'*) to describe the motion behaviour of a moving object. Some examples of the moving features are presented in the Table IV.

Speed: The speed of a trajectory segment $\overline{TS}$ is one of the most important moving features. For instance, if the speed of $\overline{TS}$ is higher or lower than the average speed of trajectories on the same road, the speed feature can significantly distinguish $\overline{TS}$ from others.

Number of Stay Points: Stay points are places where the moving object stays for a long time. The occurrence of stay point is usually caused by traffic lights or some contingency events, such as traffic jam, temporal parking for buying a newspaper, etc. If the number of stay points is unusually large, $\overline{TS}$ may have a very different moving pattern compared with other trajectories.

Number of U-turn: A U-turn is a sharp directional change of the moving object, which is usually abnormal compared with other trajectories. In real life, people often make a U-turn when they realize they are moving in wrong direction or have missed the destination. Some U-turns may violate the traffic rules or cause traffic hazard, therefore should be highlighted in the summary.

It is worth noting that the above routing and moving features are just examples to demonstrate the feasibility of our prototype STMaker. Many other possible features can be added into our system as required by applications, which is discussed in Sec. VI-B.

IV. TRAJECTORY PARTITION

In this section, we introduce the trajectory partition algorithm in STMaker. We first give a formal definition of a *trajectory partition* as follows:

Definition 5 (Trajectory Partition): A partition of a symbolic trajectory $\overline{T}$ is $\mathbb{P}_{\overline{T}}$ such that

- Each partition $\overline{TP} \in \mathbb{P}_{\overline{T}}$ is a sub-trajectory of $\overline{T}$ made up of contiguous trajectory segments, i.e.,

$$\overline{TP} = [\overline{TS}_i, \overline{TS}_{i+1}, \cdots, \overline{TS}_{i+j}]$$

- $\bigcup_{\overline{TP} \in \mathbb{P}_{\overline{T}}} \overline{TP} = \overline{T}$
- $\overline{TP}_i \cap \overline{TP}_j = \emptyset, \forall i, j.$

Clearly, each segment in the symbolic trajectory is covered by exactly one partition, and thus will be guaranteed to be summarized exactly once in the description.

Although any partition $\mathbb{P}_{\overline{T}}$ of a trajectory can lead to a summary, not all of them are suitable for a good one. **First of all, it is better for each partition to have its source and destination well-known, or more formally, significant.** For example, the description of a partition starting from the Times Square is more understandable to people than that starting from the National Hockey League building, which is only 300 meters away from the Times Square. **Second, it is easier to generate more compact summaries if the trajectory segments within the same partition are of similar features.** For instance, if the moving speed varies significantly within a partition, it is difficult to summarize the driving behavior of this partition using a few words.

A. Solution Overview

Different from previous trajectory partition algorithms which partition trajectories into pieces according to time interval, stops and etc., our partitioning is according to features which measure the travel behaviors. Based on this intuition, we propose a trajectory partition algorithm by leveraging the power of Conditional Random Field (CRF). CRF is a popular model used for image segmentation in computer vision, where pixels in an image are assigned into several regions according to their similarities, e.g., pixels representing sky, river or grass. Inspired by this, we model the trajectory partition problem as a process of labeling each trajectory segment $\overline{TS}_i$ with a tag, which satisfies the following two requirements: (1) There are at most $|\overline{T}| - 1$ different tags in total; (2) If two trajectory segments are labeled with the same tag t, then all the trajectory segments in between must be labeled by t.

We model a symbolic trajectory as an undirected graph $G(V, E)$, where each trajectory segment $\overline{TS}_i$ is a node in V, and for each pair of consecutive segments $\overline{TS}_i$ and $\overline{TS}_{i+1}$ there is an edge in E connecting their corresponding nodes. Each $\overline{TS}_i \in V$ is associated with an random variable $\mathbb{X}_i$, denoting its tag. For simplicity, we assume a common state space $\mathfrak{X}$ for all random variables $\mathbb{X}_i$, such that $\mathfrak{X} = \{1, \cdots, |\overline{T}| - 1\}$. On G, we define a clique system $\mathfrak{C} = \{C_i, i = 1, \cdots, |\overline{T}|\}$, where C_i contains two nodes $\overline{TS}_i$ and $\overline{TS}_{i+1}$. We formally define the CRF model on G and $\mathfrak{C}$ as follows, which provides a probabilistic framework for calculating the probability of the label sequence $\mathbb{X}$, i.e., $[\mathbb{X}_1, \mathbb{X}_2, \cdots, \mathbb{X}_{|\overline{T}|}]$, globally conditioned on $\overline{T}$:

$$\begin{aligned}\Pr(\mathbb{X}|\overline{T}) &= \frac{1}{Z} \exp\{-\sum_{C \in \mathfrak{C}} \Phi_C(\mathbb{X})\} \\ &= \frac{1}{Z} \exp\{-\sum_{i=1}^{|\overline{T}|-1} \Phi(\mathbb{X}_i, \mathbb{X}_{i+1}, \overline{TS}_i, \overline{TS}_{i+1})\} \quad (1)\end{aligned}$$

where Z is the normalization constant that makes the probability of all state sequences sum to one:

$$Z = \sum_{\mathbb{X}_i \in \mathfrak{X}} \exp\{-\sum_{i=1}^{|\overline{T}|-1} \Phi(\mathbb{X}_i, \mathbb{X}_{i+1}, \overline{TS}_i, \overline{TS}_{i+1})\}$$

In order to find the best label sequence $\mathbb{X}_{opt}$, we need to maximize the probability $\Pr(\mathbb{X}|\overline{T})$. In other words we need to minimize the sum of $\Phi(\mathbb{X}_i, \mathbb{X}_{i+1}, \overline{TS}_i, \overline{TS}_{i+1})$. In the next subsection, we will define $\Phi(\mathbb{X}_i, \mathbb{X}_{i+1}, \overline{TS}_i, \overline{TS}_{i+1})$, which encodes the relationship between the tags $\mathbb{X}_i$, $\mathbb{X}_{i+1}$ of two consecutive trajectory segments $\overline{TS}_i$, $\overline{TS}_{i+1}$.

B. Potential Function $\Phi(\mathbb{X}_i, \mathbb{X}_{i+1}, \overline{TS}_i, \overline{TS}_{i+1})$

Recall the two guidelines (in bold) of how to conduct a good partition in Section IV. Now we translate them into the following formal expressions: *If two contiguous trajectory partitions $\overline{TS}_i$ and $\overline{TS}_{i+1}$ are labeled with different tags, the significance $l_i.s$ of l_i should be high; if two trajectory segments $\overline{TS}_i$ and $\overline{TS}_{i+1}$ are labeled with the same tag, the similarity $S(\overline{TS}_i, \overline{TS}_{i+1})$, which measures the similarity of $\overline{TS}_i$'s and $\overline{TS}_{i+1}$'s various features, should be high.* Thus, $\Phi(\mathbb{X}_i, \mathbb{X}_{i+1}, \overline{TS}_i, \overline{TS}_{i+1})$ is defined as follows:

$$\Phi(\mathbb{X}_i, \mathbb{X}_{i+1}, \overline{TS}_i, \overline{TS}_{i+1}) = \begin{cases} -S(\overline{TS}_i, \overline{TS}_{i+1}) \ , & \text{if } \mathbb{X}_i = \mathbb{X}_{i+1} \\ -C_a \cdot l_i.s \ , & \text{if } \mathbb{X}_i \neq \mathbb{X}_{i+1} \end{cases} \quad (2)$$

where C_a is a positive constant specified by users, reflecting the importance of the significance of l_i.

Landmark significance $l.s$ is used to measure the familiarity of the landmark l to average people. To measure the familiarity of landmarks, in this work we utilize the online check-in records from a popular location-based social network (LBSN) and trajectories of cars in the target city, as these two datasets are large enough to cover most areas of the city. We leverage a HITS-like algorithm [41] to infer the significance of landmarks, by modeling the travellers as authorities, landmarks as hubs, and check-ins/visits as hyperlinks.

Next, we introduce the similarity measure $S(\overline{TS}_i, \overline{TS}_{i+1})$ between two consecutive segments. Traditional trajectory similarity/distance measures, e.g., Euclidean distance and LCSS, directly use the latitude, longitude and time-stamp of the raw trajectory to measure the spatial or spatial-temporal similarity/distance between two trajectories. Instead of spatial-temporal similarity, we will measure their similarity in travel behaviours, i.e., features. Recall that we extract mainly two types features about each segment, namely routing feature and moving feature. The similarity measure $S(\overline{TS}_i, \overline{TS}_{i+1})$ will incorporate both of them.

In order to measure the similarity of all these features of two trajectory segments, each feature should be comparable. Thus, we normalize each feature f of $\overline{TS}_i$ to a value ranging from 0 to 1. The normalizing constant of f is the biggest feature value among all the trajectory segments of $\overline{T}$. After normalization, all the features $\mathbb{F}$ of a trajectory segment $\overline{TS}_i$ form a $|\mathbb{F}|$-dimension vector $\vec{v}_i$. Therefore, measuring the similarity $S(\overline{TS}_i, \overline{TS}_{i+1})$ of two continuous trajectory segments is to measure the similarity of two vectors. We employ the most widely used vector similarity measure–Cosine Similarity [29] as our similarity measure. Since different people have different interest in different features (e.g., one may have higher interest in 'speed' feature), STMaker allows the user to specify the weight of each feature, we denote the feature weight of f by w. The bigger w is, the higher the probability for the trajectory segments, with higher similarity of f, to fall in a same partition. All the feature weight w forms a $|\mathbb{F}|$-dimension weight vector $\vec{w}$, where $\vec{w}_j$ stands for the weight of feature f_j. Using these two vectors, $S(\overline{TS}_i, \overline{TS}_{i+1})$ is defined as following:

$$S(\overline{TS}_i, \overline{TS}_{i+1}) = \frac{1}{2} \cdot \left(\frac{\sum_{j=1}^{|\mathbb{F}|} w_j \cdot \vec{u}_j \cdot \vec{v}_j}{\sqrt{\sum_{j=1}^{|\mathbb{F}|} w_j \cdot \vec{u}_j^2} \cdot \sqrt{\sum_{j=1}^{|\mathbb{F}|} w_j \cdot \vec{v}_j^2}} + 1 \right) \quad (3)$$

where the $\vec{u}$ and $\vec{v}$ are the feature vectors of $\overline{TS}_i$ and $\overline{TS}_{i+1}$ respectively; u_j and v_j are the j's dimension of u and v respectively. Note that $S(\overline{TS}_i, \overline{TS}_{i+1})$ ranges from 0 to 1.

C. Finding Optimal Partition

In our neighborhood system $\mathfrak{C}$, each variable $\mathbb{X}_i$ is only directly coupled with $\mathbb{X}_{i-1}$ and $\mathbb{X}_{i+1}$. Therefore, the CRF model is defined on a chain-like graph. Optimizing Equation (1) is a maximum a posteriori probability (MAP) problem, and thus dynamic programming (DP) can be applied to solve the MAP.

We define the DP state as (i) which represents the score of the potential function Φ on the first i trajectory segments. The state transition function is defined as

$$(i) = \min \begin{cases} (i-1) - C_a \cdot l_i.s \\ (i-1) - S(\overline{TS}_{i-1}, \overline{TS}_i) \end{cases} \quad (4)$$

The initial state is that $(1) = 0$. The final partition result is $(|\overline{T}|)$. The global optimal trajectory partition $\mathbb{X}_{opt}$ is used as the default partition in STMaker.

D. Finding Optimal k-Partition

Nevertheless, different people have different requirements of the summarization granularity. This granularity can be directly reflected in how fine-grained the trajectory is partitioned and described. To be specific, for example in the coarsest case, the whole trajectory is treated as a single piece, and only the most significant features can be described in the summary; on the contrary, in a more detailed case, the whole trajectory can be divided into several pieces where each piece has similar features and gets described in the summary. Therefore, STMaker supports the user to specify their own preference of the granularity of the summary. That is, the user specifies the number of partitions k that the trajectory is partitioned into and descriptions are generated about.

To solve this k-partition, similarly we define the DP state as a pair (i, j) which represents the score of the potential function Φ on the first i trajectory segments if the i segments are partitioned into j partitions. The state transition function is defined as

$$(i,j) = \min \begin{cases} (i-1, j-1) - C_a \cdot l_i.s \\ (i-1, j) - S(\overline{TS}_{i-1}, \overline{TS}_i) \end{cases} \quad (5)$$

Algorithm 1 illustrates the main structure of our finding optimal k-partition method. The initial state is that $(1, 1) = 0$ (line 3) while $(1, j) = \infty$ for $j > 1$. The initialization of state (i, i), which represents each of the i segments is a partition, is demonstrated by line 4 - 5. And initialization of state $i, 0$, which represents all the i segments belong to a same partition, is illustrated by line 6 - 7. The final k-partition result is (n, k) (line 12).

Algorithm 1: Finding Optimal k-Partition

```
   Input: Calibrated trajectory $\overline{T}$ and its segments $\overline{TS}$, landmark
          significance $l.s$, feature weight $w$, feature value
          $f(\overline{TS})$, the partition size $k$
   Output: The best $k$-partition of trajectory segments
1  $n \leftarrow$ the number of segments $|\overline{T}|$
2  create an array $E[0,...,n-1][0,...,k-1]$
3  $E[0][0] \leftarrow 0$
4  for $i = 1 \leftarrow n-1$ do
5      $E[i][i] = E[i-1][i-1] - C_a \cdot l_i.s$
6  for $i = 1 \leftarrow n-1$ do
7      $E[i][0] = E[i-1][0] - S(\overline{TS}_{i-1}, \overline{TS}_i)$
8  for $i = 1 \leftarrow n-1$ do
9      for $j = 1 \leftarrow i-1$ do
10         $E[i][j] = \min\{E[i-1][j-1] - c_a \cdot l_i.s,$
11                 $E[i-1][j] - S(\overline{TS}_{i-1}, \overline{TS}_i)\}$
12 return $E[n-1][k-1]$
```

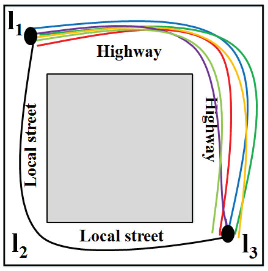
V. Feature Selection

Summarizing a trajectory partition is a process of describing the key characteristics of the routing and moving features of each partition. However, the summarization text is better to be concise for human to digest, which obviously cannot cover all the routing and moving features. For example, most roads are two-way, and people usually assume an unknown road as two-way by default. Therefore, for a trajectory partition with all the covered roads being two-way, it is unnecessary to emphasize the routing feature 'two-way' in the summary. Hence, a feature should be covered only if its value is different from normal. In other words, the selected features to be covered should be the most *irregular* features. By this strategy, the generated summary is concise, representative and can easily distinguish the given trajectory with others. In this section, we will detail how to select features to describe a given trajectory partition by leveraging historical trajectories.

In the following, we will discuss how to measure the irregular rate $\Gamma_f(\overline{TP})$ of a feature f within a partition $\overline{TP}$. Only features with higher irregular rate than a user specified threshold η will be covered in the summary.

A. Irregular Rate of Routing Features

Recall that the routing features describe where the moving object travels. Thus the irregular rate of routing features indicates how different the route of this particular trajectory partition is as opposed to the most common route of historical trajectories. Formally, given a trajectory partition $\overline{TP} = [\overline{TS}_i, \overline{TS}_{i+1}, \cdots, \overline{TS}_{i+j-1}]$ connecting l_i and l_{i+j}. We denote the most popular historical route from l_i to l_{i+j} by $\overline{PR}$. The algorithm proposed in [7] can be used to mine $\overline{PR}$ from



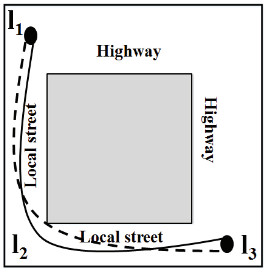


Fig. 4. Two trajectories connecting l_1 and l_3

Fig. 5. Trajectories connecting l_1 and l_3 in the road network

historical trajectories. If $\overline{TP}$ and $\overline{PR}$ are the same, there is no need to describe the routing features in summary. On the contrary, if $\overline{TP}$ is different from $\overline{PR}$, the most significant difference between $\overline{TP}$ and $\overline{PR}$ should be emphasized in the summary. For example, in Fig. V-A the given trajectory partition from l_1 to l_3 is represented by the black line and the popular route between l_1 and l_3 is represented by the coloured lines. In terms of the 'grade of road' dimension, the popular route is high way whereas the given partition is not. Obviously the 'grade of road' feature should be described in summary. Thus, we need to measure the difference/irregular rate between $\overline{TP}$ and $\overline{PR}$ in each routing feature dimension.

Given a trajectory partition $\overline{TP}$, we define its *normalized feature sequence* $F_{\overline{TP}}$ of routing feature f as $[norm(f(\overline{TS}_i)), norm(f(\overline{TS}_{i+1})), \cdots, norm(f(\overline{TS}_{i+j-1}))]$ where $norm(\cdot)$ returns the value of $\cdot$ normalized by the biggest feature value in the feature sequence. Measuring the irregular rate of $\overline{TP}$ in routing feature f can be transformed to measure the distance between $F_{\overline{TP}}$ and $F_{\overline{PR}}$, for which we design a edit-distance-like algorithm to measure.

$$d(F_{\overline{TP}}, F_{\overline{PR}}) = \begin{cases} length(F_{\overline{TP}}), & \text{if } F_{\overline{PR}} \text{ is empty} \\ length(F_{\overline{PR}}), & \text{if } F_{\overline{TP}} \text{ is empty} \\ min\{d(rest(F_{\overline{TP}}), rest(F_{\overline{PR}})) + \\ \quad cost(head(F_{\overline{TP}}), head(F_{\overline{PR}})), \\ \quad d(rest(F_{\overline{TP}}), F_{\overline{PR}}) + 1, \\ \quad d(F_{\overline{TP}}, rest(F_{\overline{PR}})) + 1\}, & \text{otherwise} \end{cases}$$

where $rest(\cdot)$ returns that tail part of a feature value sequence which consists of all but the first feature value, whereas $head(\cdot)$ returns the first feature value. For numerical routing feature f,

$$cost(head(F_{\overline{TP}}), head(F_{\overline{PR}})) = |head(F_{\overline{TP}}) - head(F_{\overline{PR}})| \tag{6}$$

whereas for categorical routing feature f,

$$cost(head(F_{\overline{TP}}), head(F_{\overline{PR}})) = \begin{cases} 1, \text{if } head(F_{\overline{TP}}) \neq head(F_{\overline{PR}}) \\ 0, \text{otherwise} \end{cases} \tag{7}$$

Finally, the irregular rate

$$\Gamma_f(\overline{TP}) = \frac{w_f \cdot d(F_{\overline{TP}}, F_{\overline{PR}})}{\max(length(F_{\overline{TP}}), length(F_{\overline{PR}}))}$$

where w_f is the user-specified feature weight.

B. Irregular Rate of Moving Features

Moving features indicate how the moving object travels. The moving pattern of a certain road is usually stable. For example, moving objects on a high way usually travels fast with no stops. Thus, if a moving object travels on the same high with several stops, it is unusual and it should be described in summary. Therefore a moving feature f's irregular rate should measure the difference between the travel patterns between each segment of the trajectory partition and that of the historical trajectories. For example, in Fig. 4, the solid line's speed-irregular-rate is derived on the basis of the differences between the travel pattern of the target trajectory and that of other trajectories (e.g., the dotted line) moving along the solid line, measured for the two segments respectively, viz. l_1 to l_2 and l_2 to l_3.

In order to measure the irregular rate of a moving feature on a trajectory partition, we first need to find out the feature's regular value on that partition. To this end, we extract the regular features values from the historical trajectories. For each moving feature f, a historical feature map, represented as a directed graph $G(V, E)$, is built to summarize feature f between two landmarks. Given a landmark set, a historical symbolic trajectory dataset and a certain moving object feature f, we can construct the historical feature map in the following steps:

1) Add each landmark in the landmark set to the vertex set V of the historical feature map.
2) Add a directed edge from l_i to l_j, denoted by $e(l_i, l_j)$, if there exists a trajectory $\overline{T}$ in the historical symbolic trajectory dataset travelling from l_i to l_j directly. We denote such a trajectory by $\overline{T}(l_i \to l_j)$.
3) Annotate each edge $e(l_i, l_j)$ with the average value of feature f of $T(l_i \to l_j)$ denoted by $r_{l_i \to l_j}$, e.g., the average sudden stop times between l_i and l_j and the average speed between l_i and l_j.

With the historical feature map, we can measure the irregular rate of f of a given trajectory partition $\overline{TP}$ by the following equation:

$$\Gamma_f(\overline{TP}) = w_f \cdot \frac{\sum_{t=i}^{i+j-1} |norm(f(\overline{TS}_t)) - norm(r_{l_t \to l_{t+1}})|}{|\overline{TP}|}$$

where w_f is the user-specified feature weight; $norm(\cdot)$ returns the normalized value of $\cdot$ and the normalization constant is the biggest feature value among all segments of the partition; $r_{l_t \to l_{t+1}}$ is the ordinary value of f of $\overline{TS}_t$, which can be easily computed from the historical feature map.

VI. Discussion

A. Summary Construction

Even with meaningful features selected, presenting them in their original numeric values is still hard to interpret by the the users, For example, given a summary saying that the average speed of the given trajectory partition is 70 km/h, the users can hardly tell whether the vehicle is fast or slow, e.g., moving in 70 km/h is pretty fast on a local road, whereas it is quite slow on a highway. Therefore, the final step in summarization is

TABLE V. TEMPLATE EXAMPLES

Feature	Template
Grade of road	through *given road type* (*road name*) while the most drivers choose *regular road type* (*road name*)
Road width	through *given road width* metres wide road while most drivers prefer **wider/narrower** roads
Traffic direction	through *given traffic direction* while most drivers prefer *regular traffic direction*
speed	with the speed of *given speed* km/h which was $|$*given speed - regular speed*$|$ km/h **faster/slower** than usual
# stay points	with *given # stay points* stay points (in total for about **time duration**)
# U-turns	with conducting *# U-turns* U-Turns at **places of U-turns**

TABLE VI. TEMPLATE EXAMPLES

Template
The car moved/started from *source* to *destination* through *road type*, with *feature template*
Then it moved from *source* to *destination* through *road type*, with *feature template*
Then it moved from *source* to *destination* smoothly.

to provide interpretable descriptions for the selected features. We define a set of phrase templates for each feature, some of which are exemplified in Table V.

In the templates, all the italics need to be replaced with the actual values of the selected features. For categorical features, the feature values are presented with the actual semantic meaning, e.g., ‘highway’ or ‘express road’ for the ‘grade of road’ feature, rather than the meaningless numbers, ‘1’ or ‘2’. For the numeric features, we further provide several intuitive descriptors (in bold font) by comparing the actual feature value with the feature’s average/regular value. As an example, the irregular ‘speed’ feature can be either faster or slower than the average speed value. Besides, feature extraction can also produce by-products, which could be very useful in the templates. For example, extracting the ‘# of stay points’ feature will also provide where the stay points take place and how long the moving object stays. Both of them can be presented in template to enrich the summary.

In order to make the summarization more fluent, we also define several sentence templates, shown in Table VI, such as ‘The car moved from *source* to *destination* through *road type*, with *feature template*. Landmarks and selected features can be embedded into these templates to generate the final summaries text.

B. Extension with New Features

In real-life application, users could easily add new features into STMaker by desire. A new feature f^* can be added with the following steps: The first step is to define the type of feature f^*, i.e., whether f^* is a routing feature or a moving feature, a numerical feature or a categorical feature. If f^* is a categorical feature, we define consecutive integers to represent the categories of f^*. The second step is to collect regular value of f^*. If f^* is a routing feature, the regular value of f^* on each road needed be collected from third-party sources, such as digital map; if f^* is a moving feature, a historical feature map of f^* need to be built with the techniques discussed in Sec. V-B. The third step is to create feature template for f^*, following the rules introduced in Sec. VI-A.

C. Text Processing

The research on text processing is very mature compared with trajectory processing. After summarizing the trajectories using text, many text processing techniques, e.g., text indexing, text clustering and text categorization, can be directly applied on the summaries. For example, applying the text clustering method on summaries of all the trajectories in a certain region at a specific time period, we can have a quick overview about the traffic condition.

VII. EXPERIMENT

In this section, we conduct extensive experiments to validate the effectiveness of our partition-and-summarization framework. Our system is implemented in Java. All the experiments are run on a computer with Intel Core i7-2600 CPU (3.40GHz) and 8 GB memory.

A. Experiment Setup

Commercial Map: We use the commercial map of a large city—Beijing—provided by a collaborating company. The commercial map is used to build the landmark dataset, and to provide routing features which are essential for our algorithm (Section III-A and V-A).

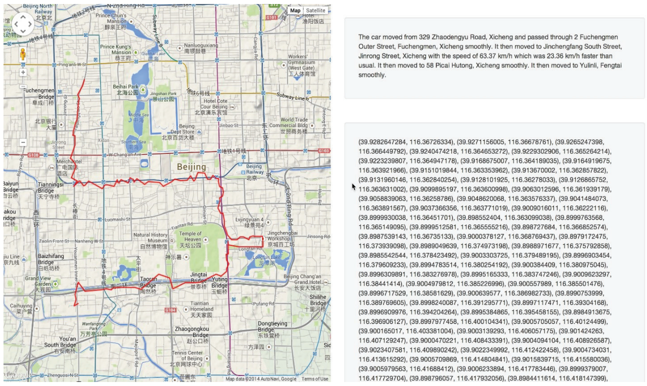
Landmark Dataset: The landmark dataset consists of two parts: the turning point dataset extracted from the commercial map, and the POI dataset of Beijing provided by a reliable third-part company. We extract about 32,000 turning points from the commercial map. The raw POI dataset has about 510,000 POI points. We cluster the raw POI dataset into approximately 17,000 clusters using DBSCAN [12], and use the geometric centers of the clusters as the landmarks.

Trajectory Dataset: We use a real-world trajectory dataset generated by 33,000+ taxis in Beijing over three months. This dataset has more than 100,000 trajectories. We randomly split the dataset into two parts: a training dataset of 50,000 trajectories, and the rest trajectories as a testing dataset. The training dataset is used to mine popular routes between the landmarks, and to build the historical feature map. The testing dataset is used to test the effectiveness of our framework. In the following, we refer to the summarization results of the testing dataset as the ***summary dataset***. Thus, there are about 50,000 summaries in summary dataset.

B. Evaluation Approach

We study both the effectiveness and efficiency of our partition-and-summarization framework. In all our algorithms, we set the weight of the landmark significance in the potential function as 0.5, the feature weight as 1 and the irregular rate threshold for a selected feature as 0.2.

We use 6 features in the experiments. including GR, RW, TD, Spe, Stay and U-turn, which represent grade of road, road width, traffic direction, speed, # stay points and # U-turn respectively. To study the effectiveness, we study the following



The car moved from 329 Zhaodengyu Road, Xicheng and passed through 2 Fuchengmen Outer Street, Fuchengmen, Xicheng smoothly. It then moved to Jinchengfang South Street, Jinrong Street, Xicheng with the speed of 63.37 km/h which was 23.36 km/h faster than usual. It then moved to 58 Picai Hutong, Xicheng smoothly. It then moved to Yulinli, Fengtai smoothly.

```
(39.9282647284, 116.36726334), (39.9271156005, 116.36678761), (39.9265247398,
116.366449792), (39.9240474218, 116.364653272), (39.9229302906, 116.365264214),
(39.9223239807, 116.364947178), (39.9168675007, 116.364189035), (39.9164919675,
116.363921966), (39.9151019844, 116.363353962), (39.913670002, 116.362857822),
(39.9131960146, 116.362840254), (39.9128101925, 116.36278033), (39.9126865752,
116.363631002), (39.9099895197, 116.363600998), (39.9063012596, 116.361939179),
(39.9058839063, 116.36258786), (39.9048620068, 116.363576337), (39.9041484073,
116.363891567), (39.9037366356, 116.363771019), (39.9009016011, 116.36222116),
(39.8999930038, 116.36451701), (39.898552404, 116.363099038), (39.8999763568,
116.365149095), (39.8999512581, 116.365555216), (39.898727684, 116.366852574),
(39.8987539143, 116.36735133), (39.9000378127, 116.368769437), (39.8979172475,
116.373939098), (39.8989049639, 116.374973198), (39.8988971677, 116.375792858),
(39.8985542544, 116.378423492), (39.9003303725, 116.379489195), (39.8996903454,
116.379609233), (39.8994783514, 116.380254192), (39.900384409, 116.380975045),
(39.8996309891, 116.383276978), (39.8995165333, 116.383747246), (39.9009623297,
116.38441414), (39.9004979812, 116.385226996), (39.900557989, 116.385501476),
(39.8996717529, 116.38581629), (39.900639577, 116.386982733), (39.8990753999,
116.389769605), (39.8998240087, 116.391295771), (39.8997117471, 116.39304168),
(39.8996909976, 116.394204264), (39.8995384865, 116.395458155), (39.8984913675,
116.396906127), (39.8997977458, 116.400104341), (39.9005705057, 116.40124499),
(39.900165017, 116.403381004), (39.9003139293, 116.406057175), (39.901424263,
116.407129247), (39.9000470221, 116.408433391), (39.9004094104, 116.408926587),
(39.9023407581, 116.409890242), (39.9022349992, 116.412422458), (39.9004734031,
116.413615292), (39.9005709869, 116.414804841), (39.9015839715, 116.415580036),
(39.9005979563, 116.41688412), (39.9006233894, 116.417783446), (39.8999379007,
116.417729704), (39.898796057, 116.417932056), (39.8984411614, 116.418147399),
```

Fig. 7. User interface of STMaker

4 aspects of the summarization: (1) whether the summary can reflect the changing of environments, e.g., time, (2) whether the landmarks mentioned by summaries are significant, (3) the impact of feature weight and partition size to our summarization framework, and (4) feedback from volunteers about whether the summary helps understanding of the trajectory. As to the efficiency test, we record the time cost for a single trajectory summarization by varying partition size k and the size of the trajectory.

C. Performance Evaluation

1) Case Study: Before conducting the quantitative performance evaluation, we show a case study of our summarization system in Fig. 6, which shows one trajectory's different summaries of different granularities. Fig. 6(a) shows the $k = 1$ summary of the trajectory that it has two stay points in the whole trajectory. Fig. 6(b) illustrates the $k = 2$ summary by dividing the trajectory into two partitions and specifies that two stay points exist in the first part and a U-turn occurs in the second part. Fig. 6(c) demonstrates the finest grained summary ($k = 3$) of the trajectory. Besides the information given by $k = 2$ summary, another significant landmark (Suzhou Road) is highlighted in the summary. We can see that more detailed information is shown with the growing of k. Also the summaries given by our system can well describe the routes as well as the moving patterns of the trajectories, which one may hardly tell directly from the map.

The user interface of STMaker is demonstrated in Figure. 7. At the lower right corner of the figure is the raw trajectory data and at the upper right corner is the summary of the trajectory. From the figure we can see that the raw trajectory is text-heavy and hard to understand while the summary gives an intuitive view of where and how the moving object travels. Also the data volume of the raw trajectory is big while the summary is more compact.

2) Effectiveness of Trajectory Summarization: **Summaries at Different Time**. The trajectories during different time of the day could be very different. For example, most trajectories during the rush hours move at low speed and have more stay points than usual. Thus, the summarization should reflect such temporal differences of trajectories. In this experiment, we evaluate how the summary contents change with time. We divide the 24 hours of the day into 12 categories, each spanning over a two-hour interval. Trajectories are classified into these 12 categories according to the time of their generation. For example, a trajectory generated at 5:00 pm belongs to the category of 16:00 - 18:00. We analyze the summarization by studying how the *feature frequencies* FF of various features change in the summaries of each category. The feature frequency FF_f of a feature f is defined as follows:

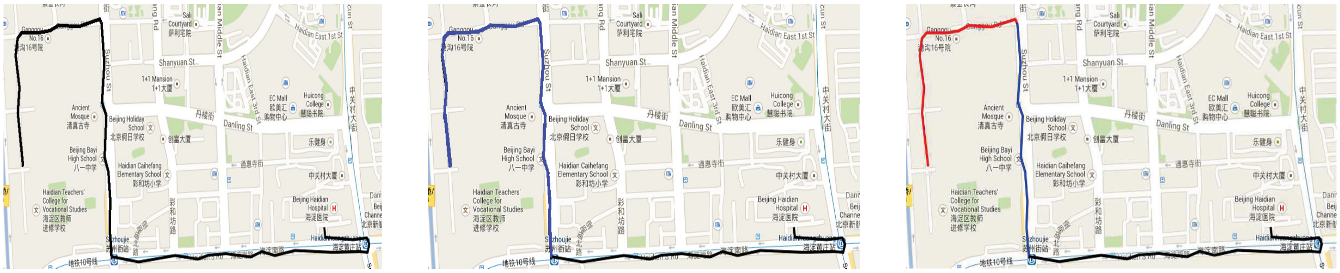
$$FF_f = \frac{\#\ summaries\ containg\ f}{\#\ total\ summaries}$$

The higher FF_f is, the more number of trajectories have irregular value on f. Notably, the weather of the eleven days is mostly sunny or cloudy, and thus most of the trajectories were collected under the similar weather. Fig. 8 shows the results of FF of all the six features we considered in our experiments.

As shown, all the features have a conspicuously higher FF during daytime (6:00 - 18:00) than those at night (18:00 - 6:00 (next day)), especially for the features: road width and speed. This contrast in the summaries clearly reflects the reported fact of heavier traffic during the daytime. If we compare the summaries during the daytime, the speed feature has a significantly higher FF in categories 6:00 - 8:00, 8:00 - 10:00, 16:00 - 18:00 and 18:00 - 20:00, which tells that more vehicles have irregular speed during these eight hours. The reason for the phenomena is that during these hours the traffic is always heavy since people need to go to work or go back home. Therefore the driving speed is slower than usual. Similar phenomena occur in relation to the features of grade of road, number of stay points and number of U-Turns. This observation agrees with our common sense that the traffic in daytime is very different from that at night. Also, observation consents to our common sense that the traffic during rush hours is very different from that during other time of the day. It verifies our expectation that our summarization framework can well reflect the change of traffic with time.

3) Effectiveness of Landmark Selection: Ideally, the starting point and destination of every trajectory partition should be of high significance, so that users can get a better idea of the trajectory. Thus, we study whether the landmarks picked in our partition step have high significance. We sort all the landmarks of the given trajectory in descending order by the landmark significance, and group them into 10 groups, i.e., top $0-10\%$, top 10%-20%, $\cdots$ landmark significance groups. For each group of landmarks, we analyze their usage frequency in the summary dataset. The usage frequencies are presented in Fig. 9. As shown, the usage frequency versus the landmark significance follows a long-tail distribution. Specifically, the landmarks in top-10%-high-significance group appear about 40% in the summary dataset, which is almost half of all the landmarks used in the summarization. Moreover, nearly 60% of the landmarks used are very popular landmarks, i.e. among the 30% groups. It proves that the landmarks we used in summaries are mostly well known to users.

4) Effect of Feature Weight and Partition Size k: In this section, we test how the feature weight w and the partition size k affects our trajectory summarization algorithm. Firstly, recall that a higher feature weight of feature f results in (1) trajectory segments with higher similarity of f have higher probability to be in a same partition; (2) feature f has higher probability to be selected in the summary. In order to verify the effect



(a) The car started from the Daoxiang Community to the Haidian Hospital with two staying points (in total for 167 seconds).

(b) The car started from the Daoxiang Community to the Suzhoujie Station with two staying points (in total for 167 seconds). Then it moved from the Suzhoujie Station to the Haidian Hospital with conducting one U-turn at Zhichun Road.

(c) The car started from the Daoxiang Community to the Suzhou Road with two staying points (in total for 167 seconds). Then it moved from the Suzhou Road to the Suzhoujie Station smoothly. Then it moved from the Suzhoujie Station to the Haidian Hospital with conducting one U-turn at Zhichun Road.

Fig. 6. Example of trajectory summarizations

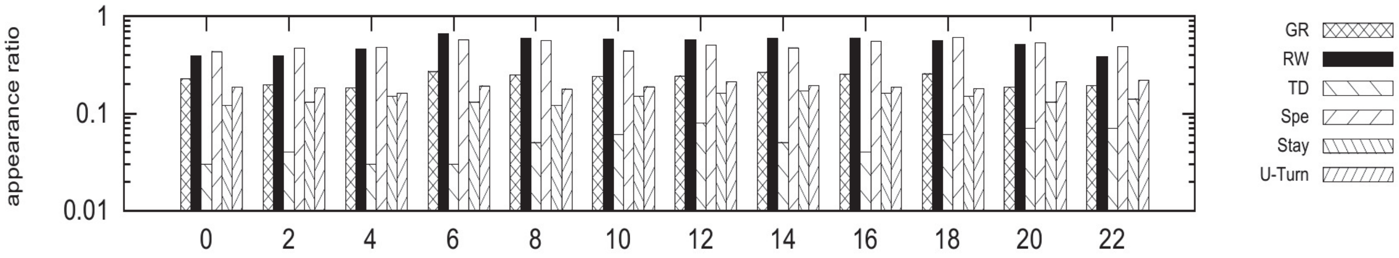


Fig. 8. Features' FF of different time

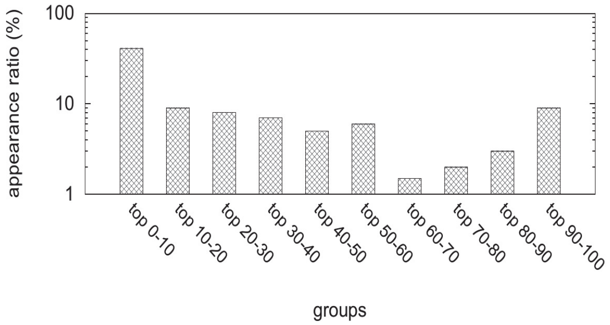
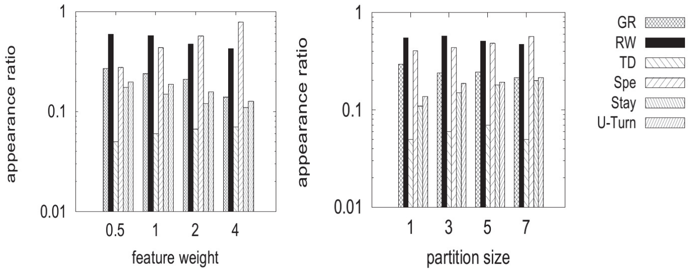


Fig. 9. usage frequency of landmark groups



(a) effect of feature weight

(b) effect of partition size

Fig. 10. Effect of parameters in trajectory summarization

of w, we conduct a controlled study by tuning w of the *Spe* feature from 0.5 to 4, while keeping w of all the other features to be default value, and summarizing randomly selected 1000 trajectories using these different weights. Fig. 10(a) shows how the eight features' FF vary with the different weights of the *Spe* feature. As shown in the figure, FF of the *Spe* feature increases gradually when the weight increases, which conforms to our expectation.

To explore the impact of partition size k, we run our summarization algorithm on 1000 randomly selected trajectories by varying k from 1 to 7, and analyze how the FF of each feature changes with k. The result is shown in Fig. 10(b). We can see that as k increases, the FF of routing features (GR, RW and TD) decrease while those of moving features (Spe, Stay, U-turn and SpeC) increase. The reason is that the larger a partition is, the more probable it is that the moving path is different from the most popular route, which in turn increases the irregular rate of routing features. Moreover, irregular moving features of the partial partition may not be significant enough for a long partition, and thus the longer the partition is, the fewer moving features are described.

5) Impressions of Users: The primary goal of trajectory summarization is to give users an intuitive view of where and how the moving object traveled. Therefore, in this experiment we test whether the users can have an intuitive view after reading the summaries. Since users' understanding is hard to quantify exactly, we propose to use four understanding levels to assess how well a user understands a trajectory after reading a summary: (1) has no idea of the trajectory; (2) has a little idea of where or how the moving object traveled; (3) has idea of where and how the moving object traveled but the summary should be improved by giving more/less information, improving the summary sentence or some other methods; (4) knows clearly where and how the moving object traveled, and the summary is well presented. We randomly select 450 summaries and ask thirty volunteer users to read fifteen summaries each. Then we ask each user to classify her understanding of the trajectory into one of the four levels. Fig. 11 shows the understanding level of the users. We can see that nearly 55% of randomly selected 450 summaries are marked at grade 4, and nearly 80% (grade 3 and 4) summaries can give users an intuitive view of the raw trajectories. This implies that the proposed trajectory summarization algorithms can achieve its primary goal.

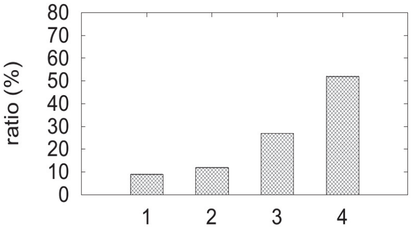
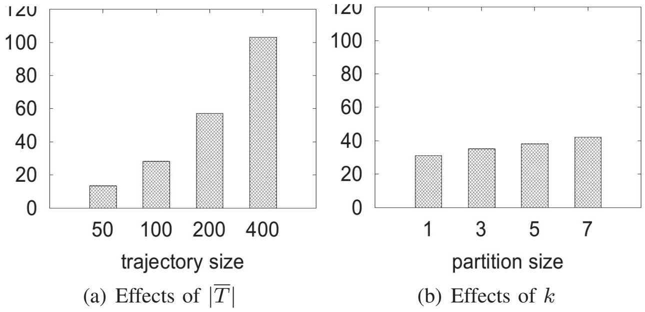


Fig. 11. User feedback

6) Summarization Time Cost: We also evaluate the time cost of our trajectory summarization algorithm, which is especially important for online summarization systems. The time cost mainly depends on the size $|\overline{T}|$ of the given trajectory T and the value of partition size k. Thus we tune $|\overline{T}|$ and k, and record the average time cost for summarizing a single trajectory. The result is shown in Fig. 12, from which we observe that most trajectories can be summarized within tens of milliseconds. With the increasing of $|\overline{T}|$ and k, the time cost increase slightly.



(a) Effects of $|\overline{T}|$

(b) Effects of k

Fig. 12. Average time cost for summarizing one trajectory

VIII. Related Work

The existing trajectory summarization works focused on how to find the most representative trajectory out of a set of trajectories. To the best of our knowledge, none of them studied using text to summarize a single trajectory. However, there are several works on summarization of text, audio and video, which share similar inspiration with our work. In this section, we review these existing trajectory summarization works, and the works on summarization of text, audio and video. We also review existing works on trajectory segmentation and trajectory annotation which are related to our partition-and-summarization framework.

Trajectory Summarization. Given a set of trajectories, [13] proposed a solution to cluster the trajectories into several groups, and represent each group by its most central trajectory. [1] summarized a set of trajectories by providing a symbolic route to represent the cardinal trajectory directions.

Text Summarization. Many works have been devoted to text summarization. [39] used Support Vector Machine to find a summary sentence in a document. However, this work is limited in that it did not leverage the relation between the sentences. [9], [21] took a different approach and made use of the relations between sentences. [28] used Conditional Random Field to partition a document and identify the summarization sentences.

Multimedia Summarization. The large amount of multimedia data available on the Internet is making audio/video summarization increasingly important. [8] exploited information retrieval techniques to summarize audio by extracting salient words from the audio. [18], [22] took a different approach which extracts non-textual features such as 'noteworthy utterances' instead of text features. [33] extracted keyframes from a video as the summarization. [18] exploited both audio and video information, and proposed the maximal marginal relevance algorithm for video summarization. [11] studied how to use natural language to summarize video, which focused on how to extract human actions from video.

Trajectory Segmentation. A few works have been conducted on trajectory segmentation. [37] proposed a method to segment heterogeneous trajectories into several parts according to different means of transportation, e.g., by bike and by car. This trajectory segmentation method can hardly be applied on a trajectory generated by the same transportation.

Trajectory Annotation. Dedicated algorithms are independently designed for trajectory annotations with geographic regions or lines. Regarding trajectory annotation with geographic regions, studies [30], [23] focused on computing topological correlations (called spatial predicates) between trajectories and regions. Regarding trajectory annotation with geographic lines, many works [2], [36], [24], [20] have been focusing on identifying the correct road segment on which a vehicle is traveling, . [2] used only geometric information of the underlying road network and apply distance measurements to generate line annotation. [36] accounted for the connectivity and contiguity of the road networks, rather than only the geometric distances. [24], [20] studied generating annotations for low-sampling-rate trajectories.

IX. Conclusions

In this paper we have taken an important step towards making sense of trajectory data by automatically generating a summaries text for individual trajectory. A partition-and-summarization framework has been proposed, which splits a trajectory into several partitions with similar travel behaviour and generates summarization to describe the most significant features for each partition. We conducted extensive experiments on a real-life trajectory dataset. The experiment results show that our summarization framework can reflect the most representative features of the trajectories. We expect this work will trigger several interesting open problems in this direction, such as summarization of trajectory group, semantic queries on trajectory summarization, etc.

Acknowledgments

This research is partially supported by the Natural Science Foundation of China (Grant No.61232006), The Australian Research Council DP (DP120102829), ARC DECRA (DE140100215). We thank Mr. Abdullah Hamid for his help in experiment analysis.

References

[1] S. Andrae and S. Winter. Summarizing gps trajectories by salient patterns. *Strobl, S., Blaschke, T. and Griesebner, G.(Edt.): Angewandte Geoinformatik*, 2005.

[2] D. Bernstein and A. Kornhauser. An introduction to map matching for personal navigation assistants. 1998.

[3] Y. Cai and R. Ng. Indexing spatio-temporal trajectories with chebyshev polynomials. In *SIGMOD*, pages 599–610, 2004.

[4] X. Cao, G. Cong, and C. S. Jensen. Mining significant semantic locations from gps data. *Proceedings of the VLDB Endowment*, 3(1-2):1009–1020, 2010.

[5] V. Chakka, A. Everspaugh, and J. Patel. Indexing large trajectory data sets with seti. In *CIDR*, 2003.

[6] L. Chen, M. Özsu, and V. Oria. Robust and fast similarity search for moving object trajectories. In *SIGMOD*, pages 491–502, 2005.

[7] Z. Chen, H. Shen, and X. Zhou. Discovering popular routes from trajectories. In *ICDE*, pages 900–911, 2011.

[8] H. Chiori and S. Furui. Speech summarization: an approach through word extraction and a method for evaluation. *IEICE TRANSACTIONS on Information and Systems*, 87(1):15–25, 2004.

[9] J. M. Conroy and D. P. O'leary. Text summarization via hidden markov models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 406–407. ACM, 2001.

[10] P. Cudre-Mauroux, E. Wu, and S. Madden. Trajstore: An adaptive storage system for very large trajectory data sets. In *ICDE*, pages 109–120, 2010.

[11] D. Ding, F. Metze, S. Rawat, P. F. Schulam, S. Burger, E. Younessian, L. Bao, M. G. Christel, and A. Hauptmann. Beyond audio and video retrieval: towards multimedia summarization. In *Proceedings of the 2nd ACM International Conference on Multimedia Retrieval*, page 2. ACM, 2012.

[12] M. Ester, H. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, pages 226–231, 1996.

[13] M. R. Evans, D. Oliver, S. Shekhar, and F. Harvey. Summarizing trajectories into k-primary corridors: a summary of results. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, pages 454–457. ACM, 2012.

[14] E. Frentzos, K. Gratsias, N. Pelekis, and Y. Theodoridis. Nearest neighbor search on moving object trajectories. In *SSTD*, pages 328–345, 2005.

[15] H. Jeung, H. Shen, and X. Zhou. Convoy queries in spatio-temporal databases. In *ICDE*, pages 1457–1459, 2008.

[16] H. Jeung, M. Yiu, X. Zhou, C. Jensen, and H. Shen. Discovery of convoys in trajectory databases. In *PVLDB*, volume 1, pages 1068–1080. VLDB Endowment, 2008.

[17] J. Lee, J. Han, and K. Whang. Trajectory clustering: a partition-and-group framework. In *SIGMOD*, pages 593–604. ACM, 2007.

[18] Y. Li and B. Merialdo. Multi-video summarization based on av-mmr. In *Content-Based Multimedia Indexing (CBMI), 2010 International Workshop on*, pages 1–6. IEEE, 2010.

[19] Z. Li, B. Ding, J. Han, and R. Kays. Swarm: Mining relaxed temporal moving object clusters. In *PVLDB*, volume 3, pages 723–734, 2010.

[20] Y. Lou, C. Zhang, Y. Zheng, X. Xie, W. Wang, and Y. Huang. Map-matching for low-sampling-rate gps trajectories. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 352–361. ACM, 2009.

[21] R. Mihalcea. Language independent extractive summarization. In *Proceedings of the ACL 2005 on Interactive poster and demonstration sessions*, pages 49–52. Association for Computational Linguistics, 2005.

[22] A. Nenkova. Summarization evaluation for text and speech: issues and approaches. In *INTERSPEECH*, 2006.

[23] M. E. Nergiz, M. Atzori, and Y. Saygin. Towards trajectory anonymization: a generalization-based approach. In *Proceedings of the SIGSPATIAL ACM GIS 2008 International Workshop on Security and Privacy in GIS and LBS*, pages 52–61. ACM, 2008.

[24] P. Newson and J. Krumm. Hidden markov map matching through noise and sparseness. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 336–343. ACM, 2009.

[25] J. Ni and C. Ravishankar. Indexing spatio-temporal trajectories with efficient polynomial approximations. *TKDE*, 19(5):663–678, 2007.

[26] D. Pfoser, C. Jensen, and Y. Theodoridis. Novel approaches to the indexing of moving object trajectories. In *VLDB*, pages 395–406, 2000.

[27] S. Šaltenis, C. S. Jensen, S. T. Leutenegger, and M. A. Lopez. *Indexing the positions of continuously moving objects*, volume 29. ACM, 2000.

[28] D. Shen, J.-T. Sun, H. Li, Q. Yang, and Z. Chen. Document summarization using conditional random fields. In *IJCAI*, volume 7, pages 2862–2867, 2007.

[29] A. Singhal. Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.*, 24(4):35–43, 2001.

[30] S. Spaccapietra, C. Parent, M. L. Damiani, J. A. de Macedo, F. Porto, and C. Vangenot. A conceptual view on trajectories. *Data & knowledge engineering*, 65(1):126–146, 2008.

[31] H. Su, K. Zheng, H. Wang, J. Huang, and X. Zhou. Calibrating trajectory data for similarity-based analysis. In *SIGMOD*, pages 833–844. ACM, 2013.

[32] H. Su, K. Zheng, K. Zeng, J. Huang, and X. Zhou. Stmaker–a system to make sense of trajectory data. *Proceedings of the VLDB Endowment*, 7(13):1701–1704, 2014.

[33] B. T. Truong and S. Venkatesh. Video abstraction: A systematic review and classification. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, 3(1):3, 2007.

[34] M. Vlachos, G. Kollios, and D. Gunopulos. Discovering similar multidimensional trajectories. In *ICDE*, pages 673–684. IEEE, 2002.

[35] H. Wang, K. Zheng, J. Xu, B. Zheng, X. Zhou, and S. Sadiq. Sharkdb: An in-memory column-oriented trajectory storage. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 1409–1418. ACM, 2014.

[36] C. E. White, D. Bernstein, and A. L. Kornhauser. Some map matching algorithms for personal navigation assistants. *Transportation Research Part C: Emerging Technologies*, 8(1):91–108, 2000.

[37] Z. Yan, D. Chakraborty, C. Parent, S. Spaccapietra, and K. Aberer. Semitri: a framework for semantic annotation of heterogeneous trajectories. In *Proceedings of the 14th international conference on extending database technology*, pages 259–270. ACM, 2011.

[38] Z. Yan, S. Spaccapietra, et al. Towards semantic trajectory data analysis: A conceptual and computational approach. In *VLDB PhD Workshop*, 2009.

[39] J.-Y. Yeh, H.-R. Ke, W.-P. Yang, I. Meng, et al. Text summarization using a trainable summarizer and latent semantic analysis. *Information Processing & Management*, 41(1):75–95, 2005.

[40] B. Zheng, K. Zheng, M. A. Sharaf, X. Zhou, and S. Sadiq. Efficient retrieval of top-k most similar users from travel smart card data. In *Mobile Data Management (MDM), 2014 IEEE 15th International Conference on*, volume 1, pages 259–268. IEEE, 2014.

[41] Y. Zheng, L. Zhang, X. Xie, and W. Ma. Mining interesting locations and travel sequences from gps trajectories. In *WWW*, pages 791–800, 2009.