STMaker–A System to Make Sense of Trajectory Data

Han Su
University of Queensland
Brisbane, Australia
h.su1@uq.edu.au

Kai Zheng
University of Queensland
Brisbane, Australia
uqkzheng@uq.edu.au

Kai Zeng
University California, Los Angeles
Los Angeles, USA
kzeng@cs.ucla.edu

Jiamin Huang
Nanjing University
Nanjing, China
hjm10@software.nju.edu.cn

Xiaofang Zhou
University of Queensland
Brisbane, Australia
uqxzhou@uq.edu.au

ABSTRACT

Widely adoption of GPS-enabled devices generates large amounts of trajectories every day. The raw trajectory data describes the movement history of moving objects by a sequence of ⟨ longitude, latitude, time-stamp ⟩ triples, which are nonintuitive for human to perceive the prominent features of the trajectory, such as where and how the moving object travels. In this demo, we present the STMaker system to help users make sense of individual trajectories. Given a trajectory, STMaker can automatically extract the significant semantic behavior of the trajectory, and summarize the behavior by a short human-readable text. In this paper, we first introduce the phrases of generating trajectory summarizations, and then show several real trajectory summarization cases.

1. INTRODUCTION

Widely adoption of GPS-enabled devices generates large amounts of trajectories every day. This inspires a tremendous amount of research effort on analyzing large scale trajectory data. Though much existing works have focused on effective indexing structures designing [6, 1], efficient query processing [9, 3] and frequent trajectory patterns mining [4, 5], few have paid attentions on semantic representation and interpretation of the trajectory data itself. Taking a look at a raw trajectory in databases, which is a sequence of triples ⟨ longitude, latitude, timestamp ⟩ as shown in Table 1, we find that this data format does not make much sense and is hard for humans to understand. In order to better understand raw trajectories, researchers have proposed semantic trajectories, which align trajectory sample points to semantic entities, i.e., roads in road networks and points of interest. Figure 3 demonstrates the idea of this approach by aligning the raw trajectory in Table 1 onto a digital map. Obviously, humans can get much better understand for where the moving object travels.

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

Table 1: Trajectory in database

Although semantic trajectories help to improve the readability of trajectory data, it still has several major disadvantages. Firstly, semantic trajectories only demonstrate the spatial information of trajectories (moving paths), while the temporal information of trajectories (moving behaviors) are not demonstrated. For example, sudden speed changes of the moving objects, which always indicate the happening of unusual things, cannot be demonstrated on map. Secondly, semantic trajectories cannot automatically highlight the "interesting" parts of the trajectories that are worth noting, such as important landmarks, major roads, etc. Although all these information have been encoded in semantic trajectories already, it needs considerable manual efforts and expertise to find them out. Thirdly, semantic trajectories are hard for communication and storage.

To address the drawbacks of semantic trajectories, we take the philosophy from text summarization in information retrieval field, and propose the STMaker system. The STMaker system uses a partition-and-summarization approach to summarize individual trajectory. The following sentence exemplifies the expected summarization for the trajectory in Table 1. *The car started from the Beijing Exhibition Center, and moved along Zizhuyuan Street passing by the Beijing Shangrila Hotel. Then it moved from the Beijing Shangrila Hotel to the Yuyuantan Park along W 3^{rd} Ring Road Middle highway, with the speed of 15 km/h which is 14 km/h slower than usual.* We can see that the textual trajectory summarizations can be superior than the raw and semantic trajectories in two aspects: (1) As the output of our framework is a summary rather than transformation of raw trajectories (e.g., semantic trajectories), data volume will be reduced significantly, which is easier to store and

Proceedings of the VLDB Endowment, Vol. 7, No. 13

communicate. (2) Despite of smaller data size, the information conveyed in the text strategically focuses on the most "interesting" parts of the trajectories, and thus makes more sense for humans.

2. SYSTEM OVERVIEW

In real life, people usually describe their trips in the following steps: (1) dividing the whole trip into several partitions with significant sources and destinations, and (2) using some significant events to specify their driving behaviors along each partition. This procedure can be demonstrated by the following tweet demonstrates: "I drove from my apartment to city through No. 5 highway slowly because of the heavy traffic, and then drove from city to the PA Hospital smoothly." In order to generate intuitive trajectory summaries, STMaker uses a partition-and-summarization approach that follows exactly the same way of how humans think. Figure 2 shows the overview of the STMaker system. Since the physical positions (latitude and longitude) of trajectories can hardly give people any intuitive view about the route of the moving object, and thus cannot serve as an description in the summary, STMaker first employs trajectory calibration [8] proposed in our previous research to rewrite the given sample-point-based trajectory into a sequence of semantic points, e.g., landmarks in the network. After the raw trajectory has been transformed to symbolic trajectory, STMaker conducts a k-partition to split the trajectory into k non-overlapping parts. During partition, the system takes considerations of multiple characteristics describing the trajectory, termed *features*, which will be discussed in Section 3. The goal of this phase is to minimize the difference between routing and moving features within the same partition, and maximize the significance of the landmarks at the two ends of this partition. Then the second phase will summarize each partition with short text. Given the fact that there are too many features to describe, we will choose the most significant features within each partition according to a novel measurement of the interestingness for each feature. In the end, the selected features will be plugged into the pre-defined phrase template to form the summary.

3. FEATURE EXTRACTION

In this section, we present the main features used by STMaker to describe the trajectories. Recalling the summary example in Section 1, we can see that there are three key elements in describing a trip: (1) landmarks to describe the source and destination (e.g., Beijing Exhibition Center), (2) where the moving object passes by (e.g., Zizhuyuan St), and (3) how the moving object travels (e.g., 14km/h slower than usual). Accordingly, we define three kinds of features to describe a trajectory.

3.1 Landmark Significance

The symbolic trajectory consists of a sequence of landmarks. It is common sense that people tend to be more familiar with the landmarks that are frequently referred to by different sources, e.g., public praise, news, bus stop, yellow pages. Here we use *landmark significance*, denoted by $l.s$, to measure the familiarity of the landmark l to average people. To infer the significance of landmarks, we utilize the online check-in records from a popular location-based social network (LBSN) and trajectories of cars in the target city,

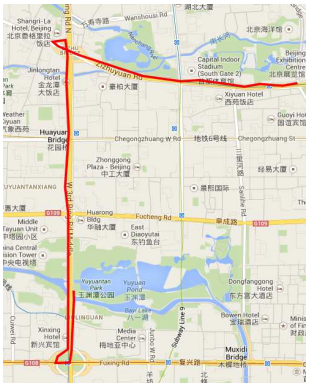

Figure 1: Trajectory shown on map

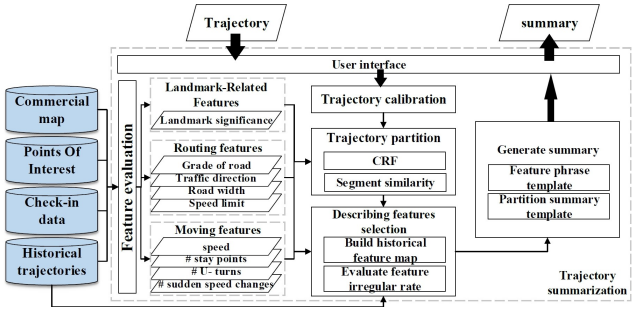

Figure 2: Framework overview

as these two datasets are large enough to cover most areas of a city. We leverage a HITS-like algorithm [10] to infer the significance of a landmark, by modeling the travellers as authorities, landmarks as hubs, and check-ins as hyperlinks.

3.2 Routing Features

Trajectory routing features indicate the characteristics related to where the moving object travels. As we focus on trajectories collected from vehicles, information about the roads they travel on are natural routing features. For example, if the whole trajectory is along a highway, then the 'moving on highway' information is important for the trajectory to be distinctive from others. More importantly, the type/features of roads 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 our prototype system we identify and use four kinds of road information (*grade of road*, *road width*, *direction* and *speed limit*) as the routing features, which can well distinguish different kinds of roads. The STMaker system extracts these features from the digital map used.

3.3 Moving Features

Moving features indicate how the moving object travels. There are many kinds of moving features, and many works have been devoted to extracting moving information from trajectories. For example, [2, 10] extract "stay points" where the sample points reported by a trajectory are in a certain region for a long time. In our system, we propose four moving features (*speed*, *number of stay points*, *number of sudden speed changes* and *number of U-turns*) to describe the motion behavior of a moving object.

4. TECHNIQUE BACKGROUND

Different people have different requirements on the summarization granularity, which indicates how fine-grained the trajectory is partitioned and described. To accommodate this requirement, we propose to partition the trajectory into k partitions according to the user's request, and summarize each partition respectively. By this way, users can tune the granularity of summary details.

4.1 Trajectory Partition

Although any k partitions of a trajectory can lead to a summary, not all of them are suitable for a good summarization. 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 with a starting point of the Times Square is more understandable to people than that with 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 a partition is made up of three segments and the moving speed varies significantly on these segments, then it is difficult to summarize the driving behavior of this partition using a concise sentence. After calibration, a trajectory is a sequence of landmarks where each landmark is assigned a time stamp. Based on this intuition, we propose a trajectory partition algorithm by leveraging the power of Conditional Random Field (CRF). The smallest sub-trajectories constructing a trajectory T, which are named as trajectory segment, are the sub-trajectories which connects two consecutive landmarks of T. Inspired by this, we model the trajectory partition problem as a process of labeling each trajectory segment with a tag, which satisfies the following two requirements: (1) There are k tags in total and all the tags should be used in the labeling process; (2) If two trajectory segments are labeled with the same tag t, then all the trajectory segments in between must be labeled by t. More formally, we associate each trajectory segment $S_i \in T$ with an random variable $\mathbb{X}_i$. The tag sequence of all segments of T is denoted by $\mathbb{X}$. The probability of $\mathbb{X}$ globally conditioned on T can be defined as:

$$\Pr(\mathbb{X}|T) = \frac{1}{Z}\exp\{-\sum_{i=1}^{|\overline{T}|-1}\Phi(\mathbb{X}_i, \mathbb{X}_{i+1}, S_i, S_{i+1})\} \quad (1)$$

where $\Phi(\mathbb{X}_i, \mathbb{X}_{i+1}, S_i, S_{i+1})$ is the potential function which models the relationship between the tags $\mathbb{X}_i$ and $\mathbb{X}_{i+1}$ of two consecutive trajectory segments S_i and S_{i+1}, and Z is a normalizing constant. In order to find the best label sequence $\mathbb{X}_{opt}$ (maximize the probability $\Pr(\mathbb{X}|\mathbb{T})$), we need to minimize the sum of $\Phi(\mathbb{X}_i, \mathbb{X}_{i+1}, S_i, S_{i+1})$.

Recall the two guidelines of how to conduct a good k partitions in beginning of Section 4.1. A reasonable definition of $\Phi(\mathbb{X}_i, \mathbb{X}_{i+1}, S_i, S_{i+1})$ is based on the following intuition: *If two trajectory partitions S_i and S_{i+1} are labeled different tags, the significance of the landmark connecting them l_{i+1} should be high; if two trajectory segments S_i and S_{i+1} are labeled the same tag, the similarity $S(S_i, S_{i+1})$, which measures the similarity of the various features of S_i and S_{i+1}, should be high.* Thus, $\Phi(\mathbb{X}_i, \mathbb{X}_{i+1}, S_i, S_{i+1})$ can be evaluated as follows:

$$\Phi(\mathbb{X}_i, \mathbb{X}_{i+1}, S_i, S_{i+1}) = \begin{cases} -S(S_i, S_{i+1}) & , \text{if } \mathbb{X}_i = \mathbb{X}_{i+1} \\ -C_a \cdot l_{i+1}.s & , \text{if } \mathbb{X}_i \neq \mathbb{X}_{i+1} \end{cases}$$

where $l_{i+1}.s$ is the significance of l_{i+1}, which is the destination of S_i and the source of S_{i+1}; C_a is a positive constant specified by users, reflecting the importance of the significance of l_{i+1}. In order to keep each segment with minimal difference in travel behaviors, $S(S_i, S_{i+1})$ will measure their similarity in routing features and moving features.

In order to measure the similarity of all these features of two trajectory segments, each feature should be comparable. Thus, we normalize each feature of S_i to a value ranging from 0 to 1. After normalization, all the features $\mathbb{F}$ of a trajectory segment S_i form a $|\mathbb{F}|$-dimension vector $\vec{v}_i$. Therefore, measuring the similarity $S(S_i, S_{i+1})$ of two continuous trajectory segments is to measure the similarity of two vectors. We employ the most widely used vector similarity measure–Cosine Similarity [7] as our similarity measure. But since different people have different interest in different features (e.g., one may have higher interest in *speed* feature), the user can specify the weight of each feature, we denote the feature weight of f by w_f. The bigger w_f is, the more important f is. All the feature weight w_f forms a $|\mathbb{F}|$-dimension weight vector $\vec{w}$, where $\vec{w}_j$ is the weight of feature f_j. Using these two vectors, $S(S_i, S_{i+1})$ is defined as following:

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

where the $\vec{u}$ and $\vec{v}$ are the feature vectors of S_i and S_{i+1} respectively. Since 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-structured 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 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

$$(i, j) = \min \begin{cases} (i-1, j-1) - C_a \cdot l_i.s \\ (i-1, j) - S(S_{i-1}, S_i) \end{cases} \quad (3)$$

The initial state is that $(1, 1) = 0$ while $(1, j) = \infty$ for $j \geq 1$. The final k partition result is given by (n, k).

4.2 Feature Selection

Summarizing a trajectory partition is a process of describing the key characteristics of each partition in dimensions of both routing and moving features. As we can see from previous sections, there are many features of a trajectory, hence not all of them should be covered in the description. Selecting the features to be covered in the summarization is the first step in the summarization process. Intuitively, the more different a feature f is from historical trajectories, the more necessary f should be mentioned in the summary. In other words, the selected features to be covered should be the most irregular features. Only features have higher irregular rate than a user specified threshold will be covered in summary.

Irregular Rate of Routing Features Recall that the routing features reflect where the moving object travels. The

(a) The car started from the China Minmetals Corportion and passed through the Capital Stadium. Then it moved from the Capital Stadium to the Beijing Shangri-la Hotel with the speed of 12.5 km/h which was 12 km/h slower than usual.

Figure 3: Example of trajectory summarizations

irregular rate of routing features should represents the difference ratio between the given trajectory route and the historical popular route. Thus, for a given trajectory segment $P = [S_i, S_{i+1}, \cdots, S_{i+j-1}]$ connecting landmarks l_i and l_{i+j}, STMaker exploits the algorithm described in [4] to identify the most popular route from l_i to l_{i+j}, which represents how the historical trajectories travel from l_i to l_{i+j}. The irregular rate is measured between the given trajectory partition and the popular route.

Irregular Rate of Moving Features Recall that the moving features describe the behaviors during the drive. Traveling on the same path, the behaviors should not vary a lot. Thus, the irregular rate of moving features should measure the behaviors' differences between the given partition and the historical trajectories traveling on the same path.

4.3 Summary Construction

As the last step, STMaker translates the values of selected highly-irregular features to readable and informative phrases. Although all the feature values are numeric, some features' numeric values do not tell any semantic meaning, i.e., *grade of road* and *traffic direction* features. So we define a set of phrase templates for each feature. For example, the template of *grade of road* feature is "through *road type* while the most drivers choose *road type*". The differences in feature values of these kind of feature should be replaced by semantic words, e.g., "highway" or "express road" of *grade of road*, rather than the meaningless numbers, "1" or "2". For features which the numeric feature value have semantic meaning, the irregular values of these features can be either bigger or smaller than the ordinary value. So we can divide irregular value of a feature into two types.

In order to give users more fluent summarization sentence, we also define several sentence templates, such as "The car moved from *source* to *destination* through *road type*, with *feature phrases*". Landmarks and selected features can be embedded into these templates to generate the final summaries text.

5. DEMONSTRATION

During the demonstration, the audience could interact with STMaker by specifying trajectories of their interest, and let the system generate text summaries. The trajectory dataset used in the demo is generated by taxis and private cars in Beijing (more than 100,000 trajectories). We get POI dataset (about 510,000 POIs) of the Beijing city from a reliable third-party company in China. We use the commercial map of Beijing provided by a collaborating company. The POIs and commercial map are used to build the landmark dataset, and to provide routing features which are essential for our algorithm (Section 3.2 and Section 4.2).

We start the demonstration by briefly explaining the work flow of our system. Then, we let our audience to pick a trajectory from the dataset or artificially draw a trajectory with specified time stamps. The audience could also designate the summary granularity by setting k, which determines how many pieces a trajectory will be partitioned. After that, STMaker will generate the summary of the given trajectory.

We show a case study of our summarization system in Figure 3. which shows an example trajectory with its corresponding summary. We can see that the summaries given by our system can well describe the routes as well as the moving patterns of the trajectories, which one can hardly tell directly from the map.

6. REFERENCES

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

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

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

[4] Z. Chen, H. Shen, et al. Discovering popular routes from trajectories. In *ICDE*, pages 900–911, 2011.

[5] W. Luo, H. Tan, et al. Finding time period-based most frequent path in big trajectory data. In *SIGMOD*, pages 195–203. ACM, 2013.

[6] D. Pfoser, C. Jensen, et al. Novel approaches to the indexing of moving object trajectories. In *VLDB*, pages 395–406, 2000.

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

[8] H. Su, K. Zheng, et al. Calibrating trajectory data for similarity-based analysis. In *SIGMOD*, pages 833–844. ACM, 2013.

[9] M. Vlachos, G. Kollios, et al. Discovering similar multidimensional trajectories. In *ICDE*, pages 673–684. IEEE, 2002.

[10] Y. Zheng, L. Zhang, et al. Mining interesting locations and travel sequences from gps trajectories. In *WWW*, pages 791–800, 2009.