Flexible Similarity Search for Enriched Trajectories

Hideaki Ohashi
Graduate School of
Informatics, Kyoto University
Yoshida Honmachi, Sakyo-ku,
Kyoto, Japan, 606-8501
Email: ohashi@db.soc.i.kyoto-u.ac.jp

Toshiyuki Shimizu
Graduate School of
Informatics, Kyoto University
Yoshida Honmachi, Sakyo-ku,
Kyoto, Japan, 606-8501
Email: tshimizu@i.kyoto-u.ac.jp

Masatoshi Yoshikawa
Graduate School of
Informatics, Kyoto University
Yoshida Honmachi, Sakyo-ku,
Kyoto, Japan, 606-8501
Email: yoshikawa@i.kyoto-u.ac.jp

***Abstract*—In this study, we focus on a method of searching for similar trajectories. In most previous works on searching for similar trajectories, only *raw* trajectory data have been used. However, to obtain deeper insights, additional time-dependent trajectory features should be utilized depending on the search intent. For instance, to identify soccer players who have similar dribbling patterns, such additional features include the correlations between players' speeds and directions. In addition, when finding similar combination plays, the additional features include the team players' movements. In this paper, we develop a framework to flexibly search for similar trajectories associated with time-dependent features, called *enriched trajectories*. In this framework, *weights*, which represent the relative importance of each feature, can be flexibly input. Moreover, to facilitate fast searching, we propose a lower bounding measure of the DTW distance between enriched trajectories. We evaluate the effectiveness of the lower bounding measure using soccer data and synthetic data. Our experimental results suggest that the proposed lower bounding measure is superior to the existing measure and works very well.**

***Keywords*—Trajectories; Similarity Search; Dynamic Time Warping.**

I. INTRODUCTION

In recent years, advances in location-acquisition techniques have resulted in the generation of many types of trajectory data such as hurricane tracking data [8], vessel motion data [12] and sports tracking data [5], [17]. Each of these datasets can be analyzed for recommendation, prediction, and event detection. A vast number of studies have introduced various analysis methods for many types of trajectories such as clustering [8] and outlier detection [7]. In this study, we focus on a search for similar trajectories, which can be applied to many of the above types of analyses.

Most methods for similar trajectory search use only *raw* trajectory data [4], [15]; however, to obtain deeper insights, additional time-dependent features should be utilized depending on the search intent. For example, if you wish to identify similar hurricanes, such additional features include the correlations between their speeds and their atmospheric pressure. In addition, if one wishes to find vessels that are moving under similar circumstances, such additional features include the weather and ocean current data. In this paper, we propose a framework to flexibly search for similar trajectories associated with time-dependent features, which we call *enriched trajectories*. In this framework, a user can flexibly set the *weights*, which represent the relative importance of each feature. Note that it is difficult to manually input all the weights accurately and obtain the desired insights without any support. Some existing works address this kind of problem. For instance, Yu et al. [18] estimate the weights between multivariate time-series from constraints, which specify what object pairs the user considers similar or dissimilar. However, solving this problem is outside the scope of this paper.

We consider that the time-dependent features of an object's trajectory can be classified into two types. One type includes the features that are derived from the primitive features of the trajectory such as the object's speed and direction. The other type encompasses the features that are obtained by combining primitive features with other data sources such as the positional relationships between the object and another object. We call the former *intrinsic features* and the latter *extrinsic features*. There is a long history of studies addressing intrinsic features [3], [9]; however, the number of studies considering extrinsic features has recently been increasing because of the increasing amount of attention being paid to cross-domain data fusion [20]. In this paper, both types of features are treated identically.

In this study, we utilize Dynamic Time Warping (DTW) [2] as a measure of the distance between enriched trajectories. DTW is a distance measure for time-series data that is simple and accurate [16]. However, the computational cost of DTW is high; therefore, many techniques have been proposed for speeding up DTW-based similarity searches. In particular, many techniques with lower bounding measures have been proposed [4], [6], [14]. Note that some people might think that intrinsic features, such as speeds and directions, are redundant because DTW can measure the distance between sequences in which they vary. However, intrinsic features are important for such cases, for example, in which we want to retrieve enriched trajectories that are spatially far away from each other but have similar speed patterns.

In this paper, to achieve fast searching, we propose a lower bounding measure that is specially designed for our proposed framework. The computational cost of this lower bounding measure is very low. We compare the performances of the proposed lower bounding measure and the existing measure for multivariate time series [11] and confirm that our proposed measure is superior to the existing measure using soccer data and synthetic data.

The contributions of this paper are as follows:

- We propose a framework for flexible similarity search for enriched trajectories considering the search intent.
- We propose a lower bounding measure that utilizes the characteristics of the proposed framework and evaluate the effectiveness.

The remainder of the paper is structured as follows: Section 2 reviews the related work. The problem statement is discussed in Section 3. In Section 4, we propose a novel lower bounding measure for fast similarity searches of enriched trajectories. We present experimental results in Section 5. In Section 6, we conclude our work with a discussion on future work.

II. RELATED WORK

In this section, we introduce the previous studies addressing intrinsic features or extrinsic features of trajectories. Unlike previous studies, our research addresses both types of features and their relative importance. Additionally, we introduce similarity measures for time-series data, including trajectory data.

A. Intrinsic Features

Pelekis et al. [9] introduced a similarity search framework for application to a trajectory database that consists of a set of distance operators based on both primitive (space and time) and derived (speed and direction) parameters of trajectories. Buchin et al. [3] defined many criteria under which a trajectory can be homogeneous, including location, heading, speed, velocity, curvature, sinuosity, and curviness, and presented a framework for segmenting a trajectory based on these criteria.

B. Extrinsic Features

Zheng et al. [19] studied the problem of efficient similarity searches for trajectories associated with activity information that is generated from location-based web applications. Zheng et al. [21] generated air quality inferences based on the trajectories of vehicles, POIs, and meteorological data.

C. Distance Measures

Dynamic Time Warping (DTW) [2] is the most widely used distance measure for time series. DTW is an algorithm that allows some points to be repeated to minimize the sum of the distance between points, which suits the unique characteristics of trajectories as follows:

- Two trajectories need not be observed synchronously for the similarity between them to be measured.
- Similar trajectory patterns often appear in different regions.

DTW is simple and accurate [16]; however, the computational cost is high. Accordingly, many attempts have been made to reduce this cost [4], [6], [10], [14].

Lee et al. [8] defined the distance function between trajectory segments as a linear combination of three components with weights. In this study, we similarly define the distance function as a linear combination with weights and utilize the weights as the relative importance of each feature.

III. PROBLEM STATEMENT

In this section, we present the problem statement and the necessary definitions. In our proposed framework, a user selects a query from a dataset of enriched trajectories and inputs weights; then, the top k results that are most similar to the query are returned. The weights represent the relative importance of features in an enriched trajectory. The definitions of enriched trajectories and the distance between them are given below.

A. Enriched Trajectory

In this study, a trajectory is defined as follows:

$$p^i = \langle p^i(1), p^i(2), \ldots, p^i(n) \rangle$$

where i is the identification index, n is the length of p^i, and the $p^i(j)$ $(j = 1, 2, \ldots, n)$ denote spatial points. We assume that all temporal intervals between adjacent points are equal in this study.

We refer to a set of spatial points and time-dependent feature values as an *enriched point*. The j-th enriched point in p^i is defined as follows:

$$ep^i(j) = (p^i(j), fv_1^i(j), \ldots, fv_m^i(j))$$

where the $fv_l^i(j)$ $(l = 1, 2, \ldots, m)$ denote time-dependent feature values and m is the number of time-dependent feature values. Thus, we represent an enriched trajectory as follows:

$$ep^i = \langle ep^i(1), ep^i(2), \ldots, ep^i(n) \rangle$$

B. Distance Between Enriched Trajectories

In this study, we utilize DTW to calculate the distance between enriched trajectories. First, we introduce the distance between two enriched points:

$$EDist(ep^i(j), ep^g(h)) \tag{1}$$
$$= w_p \cdot D_p(p^i(j), p^g(h)) + \sum_{l=1}^{m} w_{fv_l} \cdot D_{fv_l}(fv_l^i(j), fv_l^g(h))$$

where $w_p, w_{fv_1}, \ldots, w_{fv_m}$ are the weights input by the user, where all weights are real numbers greater than 0; D_p denotes the distance function for spatial points; and $D_{fv_1}, \ldots, D_{fv_m}$ denote the distance function for time-dependent feature values.

Next, we present the DTW algorithm. Let $p^i = \langle p^i(1), p^i(2), \ldots, p^i(s) \rangle$, and $p^g = \langle p^g(1), p^g(2) \ldots, p^g(t) \rangle$. DTW is defined as follows:

$$DTW(p^i, p^g) = f(s, t)$$
$$f(j, h) = D(j, h) + min \begin{cases} f(j-1, h) \\ f(j, h-1) \\ f(j-1, h-1) \end{cases}$$
$$f(0, 0) = 0, f(j, 0) = f(0, h) = \infty$$
$$(j = 1, 2, \ldots, s; h = 1, 2, \ldots, t),$$

where $D(j, h)$ denotes the distance between $p^i(j)$ and $p^g(h)$. DTW is an algorithm that allows some points to be repeated to achieve the best alignment. When we compute $DTW(ep^i, ep^g)$, $D(j, h)$ is equal to $EDist(ep^i(j), ep^g(h))$ (Equation 1).

$p^1(4)$	4	1	7	6
$p^1(3)$	1	3	5	2
$p^1(2)$	3	5	4	3
$p^1(1)$	1	2	11	3
	$p^2(1)$	$p^2(2)$	$p^2(3)$	$p^2(4)$

(a) The distance matrix and warping path between trajectories (p^1, p^2)

$fv_1^1(4)$	12	6	9	6
$fv_1^1(3)$	8	5	14	4
$fv_1^1(2)$	1	2	12	3
$fv_1^1(1)$	1	3	6	6
	$fv_1^2(1)$	$fv_1^2(2)$	$fv_1^2(3)$	$fv_1^2(4)$

(b) The distance matrix and warping path between time-dependent feature sequences (fv_1^1, fv_1^2).

$ep^1(4)$	8	3.5	8	6
$ep^1(3)$	4.5	4	9.5	3
$ep^1(2)$	2	3.5	8	3
$ep^1(1)$	1	2.5	8.5	4.5
	$ep^2(1)$	$ep^2(2)$	$ep^2(3)$	$ep^2(4)$

(c) The distance matrix and warping path between enriched trajectories (ep^1, ep^2).

Fig. 1. An example of the distance matrices and their warping paths

IV. Proposed Method

In our proposed framework, a user can obtain his desired results by modifying the specified weights. However, the distances between enriched trajectories must be recomputed whenever the user changes the weights. Because of the high computing cost of DTW, each such re-computation requires too much time. Thus, we propose a lower bounding measure that is specially designed for our proposed framework. The cost of computing this lower bounding measure is very low. Moreover, we propose a search algorithm based on this lower bounding measure.

A. Lower Bounding Measure

We propose a lower bounding measure based on the following theorem.

Theorem 1: Let $w_p, w_{fv_1}, w_{fv_2}, \ldots, w_{fv_m}$ be the weights used to compute the distance between enriched trajectories. Then,

$$DTW(ep^i, ep^g) \geq w_p \cdot DTW(p^i, p^g) + \sum_{l=1}^{m} w_{fv_l} \cdot DTW(fv_l^i, fv_l^g)$$

Theorem 1 states that a lower bound on the DTW distance between two enriched trajectories can be obtained as a linear combination of the DTW distance between the two corresponding trajectories and the DTW distances between the corresponding time-dependent feature sequences.

Before proving theorem 1, we check its validity through an example. For simplicity, we consider a pair of enriched trajectories, i.e., trajectories associated with time-dependent feature sequences, such that $(w_p, w_{fv_1}) = (0.5, 0.5)$. Figure 1 depicts the distance matrix between p^1 and p^2, that between fv_1^1 and fv_1^2 and that between ep^1 and ep^2. In addition, each red path in Figure 1 indicates an optimal warping path that represents a DTW alignment between the two sequences. In this example, the DTW distance between the trajectories is 15, that between the time-dependent feature sequences is 22 and that between the enriched trajectories is 20. Thus, it can be observed that $20 \geq 0.5 \cdot 15 + 0.5 \cdot 22$ satisfies Theorem 1.

Proof: Let $WP_{E1 \in \{ep, p, fv_1, \ldots, fv_m\}}$ be the warping path in *E1*'s distance matrix, and let $WDist_{E2 \in \{ep, p, fv_1, \ldots, fv_m\}}(WP_{E1})$ be the sum of the costs of *E2*'s grid cells on WP_{E1}. DTW chooses a path that minimizes the sum of the costs of *E1*'s grid cells; therefore, the following inequalities are satisfied.

$$\begin{aligned} &WDist_p(WP_{ep}) \geq WDist_p(WP_p) \\ &WDist_{fv_1}(WP_{ep}) \geq WDist_{fv_1}(WP_{fv_1}) \\ &\qquad\vdots \\ &WDist_{fv_m}(WP_{ep}) \geq WDist_{fv_m}(WP_{fv_m}) \end{aligned}$$

Thus, from the above inequalities and Equation 1,

$$\begin{aligned} &DTW(ep^i, ep^g) \\ &= WDist_{ep}(WP_{ep}) \\ &= w_p \cdot WDist_p(WP_{ep}) + \sum_{l=1}^{m} w_{fv_l} \cdot WDist_{fv_l}(WP_{ep}) \\ &\geq w_p \cdot WDist_p(WP_p) + \sum_{l=1}^{m} w_{fv_l} \cdot WDist_{fv_l}(WP_{fv_l}) \\ &= w_p \cdot DTW(p^i, p^g) + \sum_{l=1}^{m} w_{fv_l} \cdot DTW(fv_l^i, fv_l^g) \end{aligned}$$

Thus, we complete the proof. ■

B. Search Algorithm

We propose the search algorithm that utilize the right-hand side of the inequality in Theorem 1 as the lower bound on the DTW distance between two enriched trajectories. The idea of the proposed search algorithm is to use the lower bounding measure to prune out candidate enriched trajectories whose lower bounds are greater than the k-th DTW distance, as shown in Algorithm 1. First, we preserve the DTW distances between all pairs of trajectories and between all pairs of time-dependent feature sequences in a data structure ($preDTW$). This procedure, called the PREPROCESS PHASE, is completed before the user inputs the desired weight and selects a query. Second, we search for the k most similar enriched trajectories using the lower bounds (SEARCH PHASE). We store the identification indices i $(i = 1, \ldots, k)$ of the k candidates and the DTW distances between ep^q and ep^i in a priority queue (PQ) ranked by the DTW distance (the pop function returns the largest element). Then, we scan the remaining candidates ep^i $(i = k+1, \ldots, n)$. During this scan, we calculate the lower bounds between ep^q and ep^i using $preDTW$ and the weights W. If a lower bound is less than the k-th DTW distance in the priority queue, then we calculate the DTW distance between ep^q and ep^i; otherwise, we prune out this calculation. Furthermore, if the DTW distance between ep^q and ep^i is greater than the k-th DTW distance in the

priority queue, we pop the top of the priority queue and push in a set consisting of i and the DTW distance between ep^q and ep^i. Finally, the indices of k enriched trajectories with the lowest DTW distances are returned. Let N be the number of candidates, and let δ be the number of times the DTW algorithm is executed after the initial k candidates are stored. Then, the number of lower bounds calculated is $N-k$, and the number of DTW executions is $k+\delta$. We can create superior search algorithms using the proposed lower bounding measure, such as an algorithm in which candidates are sorted by the lower bound distance. However, we omit them due to the limit of this paper.

Algorithm 1 Enriched Trajectory Top k Similarity Search

INPUT: $\{ep^1, \ldots, ep^N\}$ ▷ $ep^i = (p^i, fv_1^i, \ldots, fv_m^i)$
INPUT: query index $q \in \{1 \ldots N\}$
INPUT: $W = \{w_p, w_{fv_1}, \ldots, w_{fv_m}\}$
INPUT: k ▷ the number of outputs
OUTPUT: the indices of the top k outputs most similar to ep^q

▷PREPROCESS PHASE
for $i \leftarrow 1$ to N **do**
  for $j \leftarrow 1$ to N **do**
    $preDTW[i][j].p \leftarrow DTW(p^i, p^j)$
    for $l \leftarrow 1$ to m **do**
      $preDTW[i][j].fv_l \leftarrow DTW(fv_l^i, fv_l^j)$
    end for
  end for
end for
▷SEARCH PHASE
for $i \leftarrow 1$ to k **do** ▷ except $i \leftarrow q$
  $PQ.push([i, DTW(ep^q, ep^i)])$
end for
for $i \leftarrow k+1$ to N **do** ▷ except $i \leftarrow q$
  $lb \leftarrow lower\_bound(preDTW[q][i], W)$
  if $lb < PQ.top.dtw$ **then**
    $true\_dist \leftarrow DTW(ep^q, ep^i)$
    if $true\_dist < PQ.top.dtw$ **then**
      $PQ.pop()$
      $PQ.push([i, DTW(ep^q, ep^i)])$
    end if
  end if
end for
return $PQ.index$

V. EXPERIMENTS

A. Experimental Setting

In this section, we evaluate the effectiveness of the proposed lower bound using a real soccer dataset and a synthetic dataset. Sports must be analyzed from various points of view; therefore, a sports dataset is a suitable means of testing our proposed flexible similarity search framework. Increasingly detailed soccer data are being collected using current technology, and analyzing these data is important for coaches, clubs and players [5]. First, we describe the brief overview of two datasets, and then, we present the time-dependent features of each dataset in detail as well as the measures of the distance between spatial points and between time-dependent feature values.

1) Datasets: In this experiment, we consider soccer player tracking data from Data Stadium Inc. This dataset consists of tracking data from 12 teams observed 25 times per second. In soccer player tracking data, each spatial point can be represented by a pair of $x^i(j)$ and $y^i(j)$ instead of by $p^i(j)$. In this experiment, we extract the FW (a forward in soccer) trajectories in the penalty area and the concurrent MF (a midfielder in soccer) trajectories from the dataset. We consider the FW trajectories to be associated with two intrinsic features, the FW's speed and direction, and two extrinsic features, the distance between the FW and the MF and the direction from the FW to the MF. The definitions of speed and direction are discussed in greater detail later. In summary, we consider enriched trajectories whose elements are as follows: ID, timestamp, $x^i(j)$, $y^i(j)$, FW speed, FW direction, distance between FW and MF, and direction from FW to MF. We utilize 9736 trajectories with an average length of 477.

In addition, we consider synthetic data whose elements are as follows: ID, timestamp, $x^i(j)$, $y^i(j)$, speed, direction and the two extrinsic features. To prepare these data, $x^i(j)$, $y^i(j)$ and the two extrinsic features were generated using the following random walk model in accordance with [14]:

$$v(i) = v(i-1) + dif(i)$$

where $v(1)$ and $dif(i)$ are uniformly distributed in the range (0, 20).

2) Speed and Direction: We denote a speed by $fv_1^i(j)$ and a direction by $fv_2^i(j)$. The speed $fv_1^i(j)$ can be defined as the distance between adjacent points if the points are sampled at equal time intervals:

$$fv_1^i(j) = \sqrt{(x^i(j+1) - x^i(j))^2 + (y^i(j+1) - y^i(j))^2}$$

Next, we define $fv_2^i(j)$ as follows:

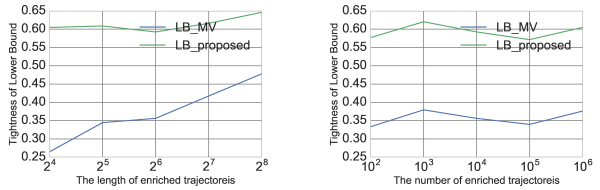
$$fv_2^i(j) = \begin{cases} \theta & (x^i(j+1) > x^i(j)) \\ \theta + \pi & (x^i(j+1) < x^i(j), y^i(j+1) \geq y^i(j)) \\ \theta - \pi & (x^i(j+1) < x^i(j), y^i(j+1) < y^i(j)) \\ \frac{\pi}{2} & (x^i(j+1) = x^i(j), y^i(j+1) > y^i(j)) \\ -\frac{\pi}{2} & (x^i(j+1) = x^i(j), y^i(j+1) < y^i(j)) \end{cases}$$

where

$$\theta = \arctan\left(\frac{y^i(j+1) - y^i(j)}{x^i(j+1) - x^i(j)}\right)$$
$$(-\frac{\pi}{2} < \arctan(x) < \frac{\pi}{2})$$

In this formulation, a direction of pure East has an angle of 0, and a direction of pure North has an angle of $+\frac{\pi}{2}$.

Note that we cannot define the speed and direction observed at the end of a trajectory $(fv_1^i(n), fv_2^i(n))$; therefore, we do



(a) Tightness when the number of trajectories is 10,000

(b) Tightness when the length of trajectories is 64

Fig. 2. Tightness of Lower Bound (weights = {1,1,1,1,1})

		Top 1		Top 10	
N	n	*without* LB(s)	LB(s)	*without* LB(s)	LB(s)
1,000	16	0.050	0.005	0.048	0.013
	32	0.161	0.028	0.162	0.057
	64	0.751	0.182	0.753	0.320
	128	2.838	0.829	2.838	1.302
10,000	16	0.410	0.027	0.411	0.052
	32	1.866	0.163	1.866	0.340
	64	7.080	0.974	7.073	1.613
100,000	16	3.899	0.198	3.890	0.262
	32	18.007	0.941	18.015	1.920
	64	71.653	7.149	71.688	11.237

TABLE I

TOP 1 AND TOP 10 RESULTS WITH RANDOMLY GENERATED WEIGHTS. NOTE THAT N MEANS THE NUMBER OF ENRICHED TRAJECTORIES AND n MEANS THE AVERAGE LENGTH OF ENRICHED TRAJECTORIES.

not consider the ends of the enriched trajectories. In addition, in the case of an unmoving object, we set the speed and direction to 0.

3) Distance between Spatial Points: Here, we define the distance between two spatial points. The Euclidean distance between spatial points has a larger range than the distance between time-dependent feature values (as defined below); therefore, we divide the Euclidean distance by $\sqrt{2}$.

$$D_p(p^i(j), p^g(h)) = \sqrt{\frac{(x^g(h) - x^i(j))^2 + (y^g(h) - y^i(j))^2}{2}}$$

Stricter adjustment of the distance scales will be a task for future work.

4) Distance between Time-dependent Feature Values: The distance between two time-dependent feature values is the absolute value of the difference between the time-dependent feature values.

$$D_{fv_l}(fv_l^i(j), fv_l^g(h)) = |fv_l^i(j) - fv_l^g(h)|$$

5) Normalization: In this experiment, we consider various types of time-dependent feature values whose ranges are distinct. For instance, the range of speeds is theoretically $[0, \infty)$, whereas the range of directions is $[-\pi, \pi]$. To obtain meaningful results, we must normalize $x^i(j)$, $y^i(j)$, and each time-dependent feature value such that they follow a distribution with an average of 0 and a variance of 1.

B. Experimental Results

We implemented the proposed algorithms in C++ and compiled them using g++ 5.2.1. The experiments were run on Ubuntu 14.04.3 with an Intel Core i7 CPU at 4.00 GHz and 32 GB of RAM. In addition, we utilized PostgreSQL 9.4.7 to store the scores computed in the PREPROCESS PHASE. We loaded all trajectory data and the data computed in the PREPROCESS PHASE into the main memory before the SEARCH PROCESS and then measured the time taken by the SEARCH PROCESS. The executable code is available at [1].

1) Comparison of the Lower Bounding Measures: We compare the performances of the proposed lower bound and LB_MV [11], which is extended LB_Keogh [6] for multivariate time series. LB_MV is inappropriate to use for multidimensional time series, thus we evaluate their performances using four time-dependent feature sequences on a random-walk dataset. To compare between our proposed lower bound and LB_MV, we use Sakoe-Chiba band [13], which is a global path constraint for DTW, and the squared difference as a measure of the distance between time-dependent feature values. In addition, we utilize *tightness of lower bound*, which is defined as the ratio of the lower bound over the true distance, as a measurement of efficiency.

$$Tightness = \frac{LowerBound}{TrueDTWDistance}$$

Tightness of lower bound is a very meaningful measure [16]. The results obtained when we set all weights to 1 and the number of trajectories to 10,000 are shown in Figure 2a. Moreover, the results obtained when we set the length of trajectories to 64 are illustrated in Figure 2b. These results suggest that our proposed lower bound is considerably superior to LB_MV. In addition, note that our proposed lower bound is simply computed based on the linear combination of each distance with weights, although it needs preparation. Therefore, the cost of calculating our proposed lower bound is much lower than that of calculating LB_MV, whose cost of computing is mn.

2) Tests on Soccer Data: We measured the time required for a top 1 similarity search for soccer enriched trajectories as described in Section V-A1. We utilized the 23 enriched trajectories corresponding to shooting FWs as queries and set all the weights equal to 1. The average computing time without the lower bound was 118.21 seconds and that with our proposed lower bound was 1.21. The computing times are discussed in greater detail below with regard to the output efficiencies observed in the subsequent studies on synthetic data. In addition, checking the usefulness of the outputs will be a task for future research.

3) Tests on Random-Walk Data: We measured the average computing times required for similarity searches on random-walk data under various conditions using 5 enriched trajectories. The experimental results obtained when we set all weights randomly, compute 100 times, and take the average of all the calculation times are shown in Table I. We measured and compared the time without the lower bound ($without\ LB$) and that with our proposed lower bound (LB). This experiment suggests that the proposed lower bounding measure is very effective for fast searches. In addition, the lower bounding measure works well especially when the number of enriched

trajectories is large, the average length of the enriched trajectories is short and the number of outputs is small.

VI. Conclusions and future work

We proposed a framework for flexible similarity searches for enriched trajectories. Moreover, we proposed a lower bounding measure and evaluate the performance under various conditions in our experiments.

There are three main prospective directions for future research as follows:

1) estimating the search intent
2) speeding up the PREPROCESS PHASE
3) searching for sub-enriched trajectories

First, the need to input all the weights imposes a high cost on users; to alleviate this, we wish to endow the framework with the ability to estimate the search intent.

Second, we need to speed up the PREPROCESS PHASE. Let m be the number of time-dependent features, let n be the average length of the enriched trajectories, and let N be the number of enriched trajectories. Then, the cost of computing the DTW distances for all features between all pairs of enriched trajectories is mn^2N^2. This indicates that the calculation costs increase dramatically as the average length and number of enriched trajectories increase. Therefore, it will be important to improve the efficiency of the PREPROCESS PHASE for practical use.

Finally, our proposed framework can identify similar overall enriched trajectories but might miss sub-enriched trajectories. In some cases, it is impossible to obtain the desired insight unless sub-enriched trajectories are considered. Therefore, it will be desirable to enhance the capability of our framework to address sub-enriched trajectories.

References

[1] Source code used in the experiment. https://drive.google.com/open?id=0B2IPZevV3CKmUGU4ZTQ5SWRramc.

[2] Berndt, D. J., and Clifford, J. Using dynamic time warping to find patterns in time series. In *KDD workshop* (1994), vol. 10, pp. 359–370.

[3] Buchin, M., Driemel, A., van Kreveld, M., and Sacristán, V. An algorithmic framework for segmenting trajectories based on spatio-temporal criteria. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems* (2010), pp. 202–211.

[4] Gong, X., Xiong, Y., Huang, W., Chen, L., Lu, Q., and Hu, Y. Fast similarity search of multi-dimensional time series via segment rotation. In *Proceedings of the Database Systems for Advanced Applications* (2015), pp. 108–124.

[5] Gudmundsson, J., and Wolle, T. Towards automated football analysis: Algorithms and data structures. In *Proceedings of the 10th Australasian conference on Mathematics and Computers in Sport* (2010).

[6] Keogh, E., and Ratanamahatana, C. A. Exact indexing of dynamic time warping. *Knowledge and information systems 7*, 3 (2005), 358–386.

[7] Lee, J.-G., Han, J., and Li, X. Trajectory outlier detection: A partition-and-detect framework. In *Proceedings of the 24th International Conference on Data Engineering* (2008), pp. 140–149.

[8] Lee, J.-G., Han, J., and Whang, K.-Y. Trajectory clustering: a partition-and-group framework. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data* (2007), pp. 593–604.

[9] Pelekis, N., Kopanakis, I., Marketos, G., Ntoutsi, I., Andrienko, G., and Theodoridis, Y. Similarity search in trajectory databases. In *Proceedings of the 14th International Symposium on Temporal Representation and Reasoning* (2007), pp. 129–140.

[10] Rakthanmanon, T., Campana, B., Mueen, A., Batista, G., Westover, B., Zhu, Q., Zakaria, J., and Keogh, E. Searching and mining trillions of time series subsequences under dynamic time warping. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining* (2012), pp. 262–270.

[11] Rath, T. M., and Manmatha, R. Lower-bounding of dynamic time warping distances for multivariate time series.

[12] Ristic, B., La Scala, B., Morelande, M., and Gordon, N. Statistical analysis of motion patterns in ais data: Anomaly detection and motion prediction. In *Proceeding of the 11th international conference on Information fusion* (2008), pp. 1–7.

[13] Sakoe, H., and Chiba, S. Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing 26*, 1 (1978), 43–49.

[14] Sakurai, Y., Yoshikawa, M., and Faloutsos, C. Ftw: fast similarity search under the time warping distance. In *Proceedings of the 24th ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems* (2005), pp. 326–337.

[15] Vlachos, M., Kollios, G., and Gunopulos, D. Discovering similar multidimensional trajectories. In *Proceedings of the 18th International Conference on Data Engineering* (2002), pp. 673–684.

[16] Wang, X., Mueen, A., Ding, H., Trajcevski, G., Scheuermann, P., and Keogh, E. Experimental comparison of representation methods and distance measures for time series data. *Data Mining and Knowledge Discovery 26*, 2 (2013), 275–309.

[17] Yajima, C., Nakanishi, Y., and Tanaka, K. Querying video data by spatio-temporal relationships of moving object traces. In *Visual and Multimedia Information Management*. Springer, 2002, pp. 357–371.

[18] Yu, W., and Gertz, M. Constraint-based learning of distance functions for object trajectories. In *International Conference on Scientific and Statistical Database Management* (2009), pp. 627–645.

[19] Zheng, K., Shang, S., Yuan, N. J., and Yang, Y. Towards efficient search for activity trajectories. In *Proceedings of the 29th International Conference on Data Engineering* (2013), pp. 230–241.

[20] Zheng, Y. Trajectory data mining: an overview. *ACM Transactions on Intelligent Systems and Technology (TIST) 6*, 3 (2015), 29.

[21] Zheng, Y., Liu, F., and Hsieh, H.-P. U-air: When urban air quality inference meets big data. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining* (2013), pp. 1436–1444.