Mining Trajectory Patterns Using Temporal Constraints

P. R.Pathare[1], Prof. S.M.Rokade[2]
[1, 2] Department of Computer Engineering
[1, 2] PREC, Loni, Maharashtra

Abstract- *Some traditional methods of UT pattern mining are inefficient as well as more complicated which only capable of identifying specific type of trajectory pattern from input dataset. We proposed UT-pattern mining framework to address the limitations of previous methods. It contains two phases to mine UT patterns such as, initial pattern mining & pattern forest construction. Clusters of various UT patterns are constructed using Traclus method. This cluster contains initial UT patterns. After pattern forest construction, patterns are classified into different categories such as, time relaxed, time constrained & time independent. Along with pattern discovery and it's classification proposed system contributes frequent UT-pattern discovery. With experimental result system demonstrates the efficiency of UT pattern framework in terms of time and memory.*

Keywords- Trajectory pattern mining, synchronous movement patterns, moving object trajectories, trajectory clustering

I. INTRODUCTION

There are two types of movement discussed in existing work such as, synchronous and asynchronous movements. Synchronous movement can be defined as, objects interacting with each others with the small time interval whereas, asynchronous patterns can be defined as, objects are moving together. Unified trajectory patterns consist of a set of trajectories which closely related to the location & time hence they are included into asynchronous category. Unifying trajectory patterns have various application areas such as, deer migration, Wolf predation on wild ungulates. UT pattern mining is very useful in learning interactions between moving objects. Previously, lots of efforts have been conducted on the work of UT pattern mining such as, flock patterns [2], convoy patterns [2], swarm patterns [4], moving clusters [8], time-relaxed trajectory joins [10], hot motion paths [5], and sub-trajectory clusters etc. From analysis of existing system it is observed that only specific trajectory pattern can be identified from given input dataset which is more tedious & inefficient task. Other limitations could be that user is unaware of which types of trajectories are hidden into given dataset. Motivating from these limitations, UT pattern mining framework is proposed to discover UT patterns from large dataset and classified them into three types of categories such as, time-relaxed, time-constrained, and time-independent patterns. Line segmentation, vector creation strategies are implemented during initial cluster creation. Whereas, in pattern forest construction, data compression, reference movement extraction, pattern distribution is performed. In pattern forest construction, dill down and roll up approaches are implemented with MDL principle.

Two types of phases are included into proposed framework such as, initial pattern discovery & granularity phase. These both phases are guided by information-theoretic formula which is based on principle of minimum description length (MDL). In sub-trajectory cluster formation two phases are included such as, partitioning phase & grouping phase. In partitioning phase, trajectory is partitioned into a set of line segment which also known as, trajectory partitioned whereas, in grouping phase similar kind of trajectories are grouped into same cluster. Grouping of line segments are based on density based clustering method. The final stage of trajectory pattern grouping is also called as, representative trajectory. It is the sequence of points similar to the ordinary trajectory generated for each cluster. Trajectory partitioned are converted to location, time space which makes analysis simpler as well as easier almost without losing an accuracy. Reference movement is calculated which capture underlying patterns of partitioned trajectory. Similarity measure is calculated between line segments is required for quantifying the degree of data compression. An approximation algorithm is used for initial UT pattern generation & it receives the set of trajectory partitioned that belonging to the same sub-trajectory cluster & returns a set of UT-patterns. In granularity phase, pattern forest is constructed using previous initial patterns.

OLAP-operations are used in pattern forest construction; it contains two types of operations such as, drill-down and roll up operations. Drill down operation decreases the cost of MDL principle by selecting dimensions automatically. Opposite to the drill down approach, roll up operation does not decrease the cost of MDL.

II. REVIEW OF LITERATURE

REMO model analysis allows inspecting relative motion of many moving bits of objects. REMO analysis have two key ideas such as, the multiple parameters represents the each motion arrange in the analysis matrix whereas, the other is spatio–temporal "behavior" and relations in each group of moving bit objects are established as patterns in the REMO matrix. Necessary patterns of relative motion i.e. constant, concurrence is actually be identified and bounded in space-time. REMO helps to find concurrences in any kind of observation data of moving point's objects. Furthermore, enhancement of model is to test and improve REMO analysis model. In testing phase, REMO model build an artificial and additional real observation data sets [1]. A geographic data mining approach detects generic aggregation patterns such as flocking behavior and convergence in geospatial lifeline data. It considers the properties of object's motion in an analytical space. [2].

A development of a generic strategy for topographical knowledge discovery in partial lifeline data is discussed in [3]. It is developed by combining crucial steps of KDD such as, reduction of data and projection, analysis of exploratory & model selection, data mining and Visualization. Football players tracked on a pitch and data points moving in an abstract ideological space used to determine generic nature of proposed approach. It extracts the set of valid, novel, useful and understandable motion patterns from these datasets.

The idea of projecting trajectories into points in higher dimensional space is feasible to discover flocks in spatio-temporal data. It is the first step towards the algorithms for finding spatio-temporal patterns, such as flocks, encounters and convergences. In preprocessing step, to reduce number of dimensions random projection is used. A tree-based algorithm is implemented for identification of flock pattern which performs very well in small numbers of time-steps the resulting running times are often very small [4]. A "Time Relaxed Spatiotemporal Trajectory Join" query is demonstrated for the symbolic join algorithm [5] but it is inefficient solution hence two heuristics solutions have represented which totally decreases query time for the TRSTJ query. Symbolic join algorithm is based on multiple origins notions that can reduce the number of false positives effectively. Another heuristic solution was based on the principle of "divide and conquer". A framework for on-line maintenance of hot motion paths is used to discover frequently traveled trails of numerous moving objects. In [6], a distributed platform with administrator and manage hotness and calculations of these paths in a spatiotemporal index, and many moving clients that issue updates only for important changes in their positions. They were focused on motion patterns that discarding obsolete paths that expire from a sliding time window. Freely moving objects are assumed by them. An observational reproduction exhibits the ability of our methodology to provide a dense representation of objects' movement, as well as its efficiency with respect to on-line maintenance of significant motion patterns.

A partition-and-group framework is utilized for trajectory clustering. It partitions trajectory patterns into a set of line segments then, combine unique line segments into a cluster. Its main advantage is to extract common sub-trajectories from a trajectory database. Based on this strategy, they have developed a trajectory clustering algorithm known as, TRACLUS algorithm. It consists of partitioning and grouping phases, it is based on minimum description length (MDL) principle [7] with some properties such as, Extensibility, Parameter Insensitivity, Efficiency, Movement Patterns and Temporal Information.

The problem of computing a longest duration flock or meeting is discussed in [8]. Several exact and approximation algorithms is provided. It has some modification as difficult as MaxClique for computations. Divide-and-conquer is performed on τ and then test same $O(n/(m\varepsilon 2))$ vertical columns. In this paper, they have determined all entities the time intervals that they are within the sweeping disk. Recursion is applied to discover a longest interval containing an at least m interval, which is done before and after $t\hat{}$ independently. Distinct from flock pattern this problem is not NP-complete and below we will give a polynomial time algorithm for fixed-meet (m, max, r), followed by a faster radius approximation algorithm. Clustering analysis on moving objects, able to provide some interesting pattern changes and is of extensive interest. Micro-cluster catches some regularity of moving objects and manages very large databases. An efficient algorithm implemented to keep moving micro-clusters graphically small. A superb clustering result could be obtained together with the knowledge about collision. A robust framework for determining a natural clustering of a given data set, based on the minimum description length (MDL) principle is proposed in [10]. The proposed framework, Robust Information-theoretic Clustering (RIC), is orthogonal to any known clustering algorithm: given a preliminary clustering, RIC purifies these clusters from noise, and adjusts the clustering's such that it simultaneously determines the most natural amount and shape (subspace) of the clusters. RIC method can be integrated with any clustering technique ranging from k-means to k-medoids. RIC framework is very flexible, with several desirable properties that previous clustering algorithms don't have. RIC framework does not strive with existing methods of clustering [10]. Framework for discovering trajectory patterns seems very useful in learning interactions between moving objects.

III. PROBLEM DEFINITION

"To mine unified trajectory patterns form given dataset and classify patterns based on time constraint and time relax manner".

IV. SYSTEM ARCHITECTURE

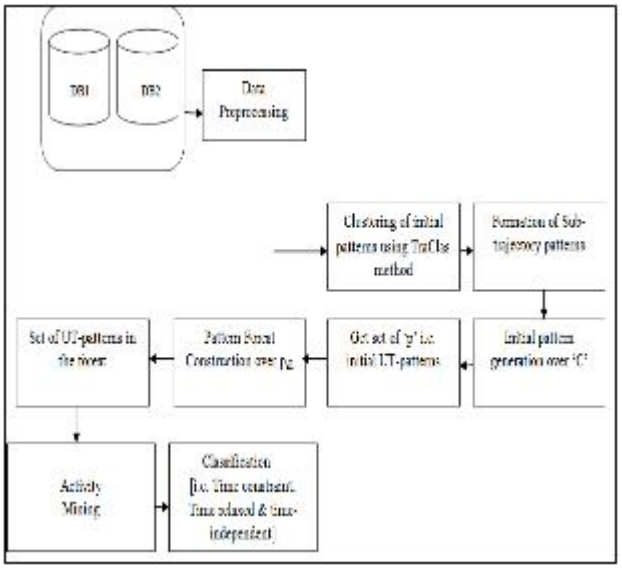

Figure 1. System Architecture

In the proposed system initially data is preprocessed and extracted required content form a given dataset. Then in initial phase it executes Traclas algorithm to find initial UT pattern set irrespective to the time constraint. Then it generates sub trajectory clusters and finds pattern forest information. After finding such patterns cluster system generates partitions of those patterns based on temporal constraint

Working flow given as following:

1. Dataset Splitting:

- In this phase, input dataset is given to the system. Dataset consists of Cattle, Deer and Elks movements.
- Dataset is nothing but huge file with finite number of these animals.

There are two main phases involved into UT pattern mining. It is given below:

1. Initial Pattern discovery
2. Pattern forest construction

1. Initial pattern discovery:

- In initial pattern discovery, sub trajectory clusters are created which contain initial UT patterns.
- For sub-trajectory cluster creation TraClas method is implemented. It is based on divide and grouping strategy.

In divide phase, each trajectory is partitioned into a set of line segments (i.e., trajectory partitions) whenever its moving direction changes rapidly.

The problem of finding the optimal set of such partitioning points is formulated by the MDL principle.

An O (n) approximate algorithm is proposed to efficiently find the near-optimal partitioning.

In grouping phase, similar types of line segments are grouped into clusters using density based clustering method. It is analogous to DBSCAN.

Final stage of the grouping phase, a model called a representative trajectory, which is a sequence of points just like an ordinary trajectory, is generated for each cluster. It is an imaginary trajectory that indicates the major movement pattern of the trajectory partitions belonging to the cluster and is obtained by calculating the average coordinates of those trajectory partitions.

2. Derived reference movement :

After identification of sub-trajectory clusters next step is to find reference movement which capture underlying patterns involved into sub-trajectory cluster. As per similarity measure, every trajectory partition is allocated to closest reference movement.

3. Analysis of UT-patterns:

A temporal cohesion is introduced to quantify temporal proximity in UT-patterns. A type of UT pattern is determined by its temporal cohesion. Temporal cohesion is based on the correlation coefficient.

4. Pattern forest construction:

From underlying pattern of trajectories, initial trajectories are captured according to MDL cost. In pattern forest construction, drill-down and roll-up approach is utilized for automatically determine location or time.

Drill-down approach :

It is used only if split decreases the MDL cost. The main purpose of drill-down approach is to derive multiple

time-constrained or smaller time-relaxed patterns from a time-relaxed pattern. Drill down approach minimizes the cost of MDL.

Roll-up approach :

- Roll up is the reverse operation of drill down approach.

V. MATHEMATICAL MODEL

'S 'is the system of UT-pattern mining such that,
S = {I, F, O}
I is the input to the system
F is system functions
O is Systems output

I: Input Parameter{I1, I2}
I1= User Login
I2= Trajectory Dataset

F:Functional Parameter {F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, F13,F14, F15, F16, F17, F18, F19,F20,F21}
F1= User login
F2= Upload trajectory dataset
F3= Create 3D- dimensional vector
F4= Define line segment
F5 Group or merge similar line segments
F6= Display initial cluster
F7= Identify average direction
F8= Find projection point
F9= Apply data compression
F10= Identify UT patterns
F11= Derive reference movement
F12= Split patterns into two section
F13= Create new initial UT patterns
F14= Construct pattern forest
F15= Apply drill down approach
F16= Apply roll up approach
F17= Create pattern forest
F18=Frequent pattern mining
F19= Create pattern set
F20= Classify patterns
F21= Generate report

O :Output Parameter{ O1, O2, O3, O4}
O1 = Initial clusters
O2= Pattern forest
O3= Frequent patterns
O4= Classified patterns

VI. ALGORITHMS

1. UT-Pattern Mine

Input: A set of Trajectories I= {TR_1, TR_2 …TR_{numtra}}
Output: A set of UT-patterns O= {UT1, UT2 …UT_{numpat}}
Processing Steps:

- Phase I-Initial pattern discovery

2. Perform sub-trajectory clustering over I based on the TRACLUS algorithm[8]

3. Get all sub-trajectory clusters C_{all}
4. For each $C \in C_{all}$ do
5. /Algorithm 2 */
6. Execute initial pattern generation over C
7. Get set P of UT-patterns as a result
8. Accumulate P into a set P_{all}
9. End for
10. Phase II-Granularity Adjustment
11. /Algorithm 3 */
12. Execute pattern forest construction over P_{all}
13. Return the set of UT patterns in the forest
14. Classify pattern into three type's i.e. time-constrained pattern, time-relaxed patterns and time independent patterns.

2. Initial Pattern Identification:

Input: A set L of trajectory partitions in a cluster C
Output: A set P of initial UT-patterns
Processing steps:

1. $L1 \leftarrow L$, $\overrightarrow{R_1} \leftarrow$ DeriveRefMovement(L1);
2. $P \leftarrow \{(\overrightarrow{R_1}, L_1)\}$;
3. Repeat
4. Choose the m^{th} UT-pattern from P, where,
   $M = \text{argmax } C(\overrightarrow{R_m}, L_m)$;
   $(\overrightarrow{R_m}, L_m) \in P$
5. Split the m^{th} UT-pattern into two splits
6. Choose the pair of trajectory partitions, where
   $(Lp,Lq) = \text{argmax dist}(Lp,Lq)$;
   $Lp,Lq \in Lm$
7. Distribute t-partitions of Lm into two
8. $L^p_m \leftarrow \emptyset$, $L^q_m \leftarrow \emptyset$,
9. For each $L_i \in$ Lm do
10. If $dist(L_i,Lp) < dist(L_i, Lq)$ then
11. $L^p_m \leftarrow L^p_m \cup \{L_i\}$;
12. Else
13. $L^q_m \leftarrow L^q_m \cup \{L_i\}$;
14. End if
15. End for
16. Derive reference movement for each split
17. $\overrightarrow{R^p_m} \leftarrow$ deriveRefMovement(L^p_m)
18. $\overrightarrow{R^q_m} \leftarrow$ deriveRefMovement(L^q_m)

19. Replace the m^{th} pattern by new ones.
20. $P' \leftarrow P-\{(Rm,Lm)\}U\{(R^p_m, L^p_m), \{(R^q_m, L^q_m)\}$
21. Check if L(H)+L(D|H) decreases
22. If MDL(P')<MDL(P) then
23. P←P'
24. End if
25. Until MDL(P')>MDL(P)
26. Return the set P of initial UT-patterns
27. Function DeriveRefMovement(L_K)
28. Consider each t-partition as a candidate
29. $R_k \leftarrow \{L|\forall L \in L_k\}$
30. Find one that minimizes the code length
31. Return s^{th} candidate $\overrightarrow{R^s_k}$, where
    $S=\text{argmin } C(R^s_k, L_k)$
    $R^s_k \in R_k$
32. End function

3. Pattern Forest Construction

Input: A set P_{all} of initial UT-patterns
Output: A pattern forest FR
Processing steps:

1. FR← P_{all} , Q← P_{all} where, Q is queue
2. Perform Drill-Down operation
3. While Q ≠Ø do
4. Pop a UT-pattern UT_i from Q
5. If UT_i can be easily spilt into UT^1_i and UT^2_i then
6. Push UT^1_i and $UT2_i$ into Q;
7. Update pattern forest
8. Add two vertexes for UT^1_i and UT^2_i into FR;
9. Add two edges for (UT_i, UT^1_i) and (UT_i, UT^2_i) into FR;
10. End if
11. End while
12. Perform Roll-up operation
13. P_c is the set of UT-pattern in the c^{th} cluster
14. $P_c \subseteq P_{all}$ do
15. For each pair of $UT_i \in P_c$ and $UT_j \in P_c$ do
16. If UT_i and UT_j merged into UT_{ij} then
17. Add UT_{ij} into P_c
18. Update pattern forest
19. Add one vertex for UT_{ij} into FR;
20. Add two edges for (UT_{ij}, UT_i) and (UT_{ij}, UT_j) into FR.
21. End if
22. End for
23. End for
24. Return the pattern forest FR.

VII. EXPERIMENTAL SETUP

As per project requirement and decided contribution JAVA platform is selected to design and develop the system. JDK1.7 is used to set up the JAVA environment. Using Eclipse (Any latest version e.g. indigo) or Netbeans system can be developed. Datasets are given as an input.

Dataset (Starkey_OR_Main_Telemetry_1993-1996_Data [12]):

Trajectory details of various animals are incorporated in the dataset. This dataset is having number of parameters like UTM Grid (East North) details, animal id, species details, date, time etc. It gives the location particular of particular animal on particular date and time. Amongst all these species, Elk (Species code is "E") and Deer (Species code is "D") is parsed. These datasets are given as an input to predict the trajectory details.

VIII. RESULT TABLE AND DISCUSSION

In proposed system, patterns are classified into two time constrained such as, time based constrained and time independent constrained. Following graph represents the time evaluation for complete processing of datasett.

Table 1: TIME EVALUATION

Dataset	**Processing time in sec.**
DEER	27.94
ELK	64.119
CATTLE	19.773

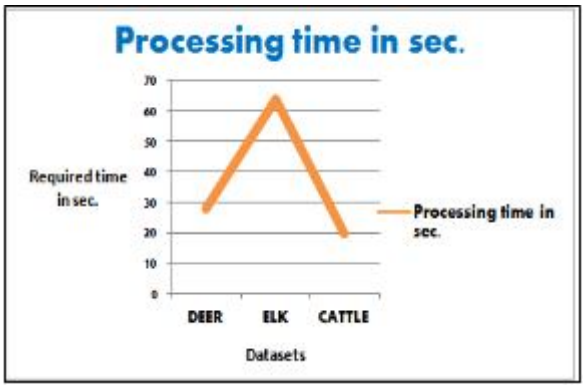

Figure 2. Graph of processing time evaluation

Figure 3, 4, and 5 represents the sample UT-patterns of deer dataset, elk dataset and cattle dataset respectively. In this Blue line represents the reference the frequent trajectories for each cluster whereas, red line represents the reference trajectory lines for each cluster.

In this each point consist of a real coordinate and timestamp. Figure 3, 4, and 5 has similar traces in the opposite direction indicated by red and blue colors.

The final output is a set of UT-patterns of the form UTi =(Ri,Li), each of which is optionally classified into one of the three types, or a set of movement paths on a map with their values of the temporal cohesion. The usefulness of the resulting UT-patterns indeed depends on the data set and application.

1. Deer Dataset:

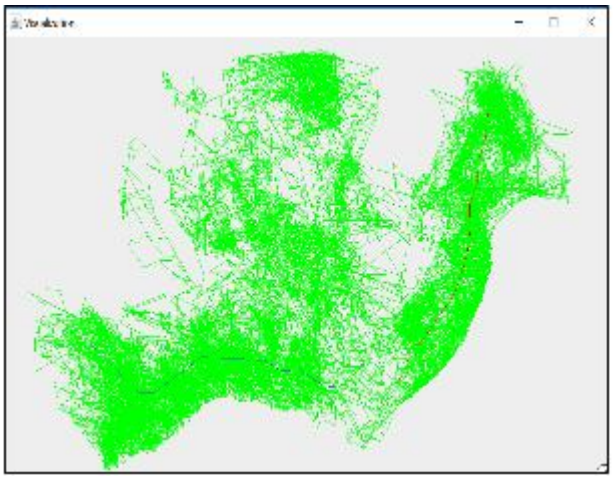

Figure 3. sample UT pattern for deer dataset

2. Elk Dataset:

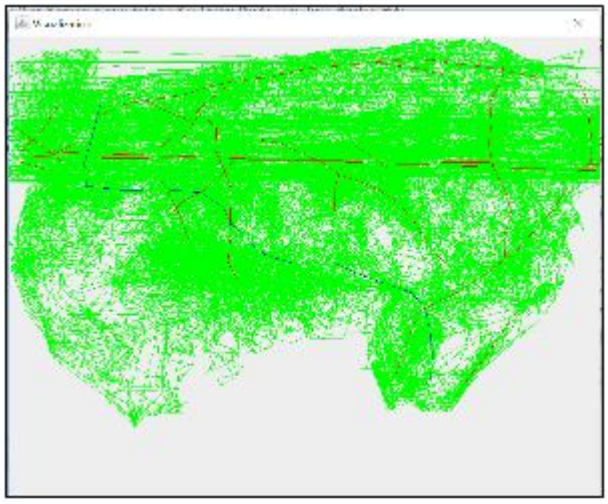

Figure 4. sample UT pattern for elk dataset

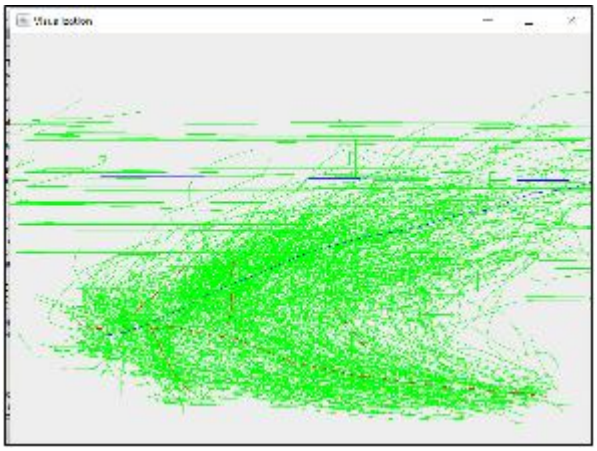

Figure 5. sample UT pattern for cattle dataset

IX. CONCLUSION

We proposed a framework of UT pattern mining. Previous methods of UT mining identify only one type of trajectory patter. To address this limitation UT pattern mining approach is implemented. There are two phase included in proposed framework such as, initial phase and granularity phase. MDL principle is described during pattern forest construction also operations like, drill down and roll up has been represented for detail level extraction of initial patterns. After pattern forest construction, patterns are classified into three types of categories such as, time-relaxed, time-constrained and time independent patterns also frequent pattern mining is takes place.

X. ACKNOWLEDGMENT

I wish to express my sincere gratitude to H.O.D Prof. S. M. Rokade of M.E. Computer Engineering Department for providing me an opportunity for presenting the topic ”Mining Trajectory Patterns Using Temporal Constraints”. I sincerely thank to my guide Prof. S.M.Rokade for his guidance and encouragement in the completion of this work.

I also wish to express my gratitude to the officials and other staff members who rendered their help during the period. Last but not least I wish to avail myself of this opportunity, to express a sense of gratitude and love to my friends and my parents for their manual support, strength, help and for everything.

REFERENCES

[1] P. Laube and S. Imfeld, “Analyzing relative motion within groups of trackable moving point objects,” in Proc. 2nd Int. Conf. Geograph. Inf. Sci., Boulder, CO, USA,

Sep. 2002, pp. 132–144.

[2] P. Laube, M. J. van Kreveld, and S. Imfeld, "Finding REMO— Detecting relative motion patterns in geospatial lifelines," in Proc. 11th Int. Symp. Spatial Data Handling, Leicester, U.K., Aug. 2004, pp. 201–214.

[3] P. Laube, S. Imfeld, and R. Weibel, "Discovering relative motion patterns in groups of moving point objects," Int. J. Geograph. Inf. Sci., vol. 19, no. 6, pp. 639–668, Jul. 2005.

[4] M. Benkert, J. Gudmundsson, F. Hubner, and T. Wolle, "Reporting flock patterns," in Proc. 14th Eur. Symp. Algorithms, Zurich, Switzerland, Mar. 2006, pp. 660–671.

[5] P. Bakalov, M. Hadjieleftheriou, and V. J. Tsotras, "Time relaxed spatiotemporal trajectory joins," in Proc. 13th ACM Int. Symp. Geograph. Inf. Syst., Bremen, Germany, Nov. 2005, pp. 182–191.

[6] D. Sacharidis, K. Patroumpas, M. Terrovitis, V. Kantere, M. Potamias, K. Mouratidis, and T. K. Sellis, "On-line discovery of hot motion paths," in Proc. 11th Int. Conf. Extending Database Technol., Nantes, France, Mar. 2008, pp. 392–403.

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

[8] J. Gudmundsson and M. J. van Kreveld, "Computing longest duration flocks in trajectory data," in Proc. 14th ACM Int. Symp. Geograph. Inf.. Syst., Arlington, VA, USA, Nov. 2006, pp. 35–42.

[9] Y. Li, J. Han, and J. Yang, "Clustering moving objects," in Proc. 10th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, Seattle, WA, USA, Aug. 2004, pp. 617–622.

[10] C. Bohm, C. Faloutsos, J.-Y. Pan, and C. Plant, "Robust informa- € tion-theoretic clustering," in Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, Philadelphia, PA, USA, Aug. 2006, pp. 65–75

[11] J. Lee, J. Han and Xiaolei Li ,"A Unifying Framework of Mining Trajectory Patterns of Various Temporal Tightness", IEEE transaction on knowledge and data mining vol,27, No.6, june 2015.