Implementation of Application - Level Semantics in Data Compression

M.Anusha1,B,Jyothi2
1 M.Tech Student, CVSR College of Engineering, Department of Computer Science, A.P. India

2 Assistant Professor of Computer Science and Engineering, Anurag group of Institutions, A.P. India

Abstract— In this paper, we move towards this direction and develop an extension of the sequential pattern mining and that analyzes the trajectories of moving objects. we first propose an efficient distributed mining algorithm to jointly identify a group of moving objects and discover their movement patterns in wireless sensor networks. Afterward, we propose a compression algorithm, called 2P2D, which exploits the obtained group movement patterns to reduce the amount of delivered data. The compression algorithm includes a sequence merge and an entropy reduction phases.

Index Terms— Data compression, distributed clustering, object tracking, Moving objects, data aggregation;

I INTRODUCTION

In this paper, we move towards this direction and develop an extension of the sequential pattern mining and that analyzes the trajectories of moving objects. we first propose an efficient distributed mining algorithm to jointly identify a group of moving objects and discover their movement patterns in wireless sensor networks. Afterward, we propose a compression algorithm, called 2P2D, which exploits the obtained group movement patterns to reduce the amount of delivered data. The compression algorithm includes a sequence merge and an entropy reduction phases. The advancements made in location acquisition expertise, such as global positioning systems and wireless sensor networks have promoted many innovative applications create a great volume of location records and consequently lead to broadcast and storing tasks. Determining the group movement patterns is more difficult than finding the patterns of a single object or all objects since recognizing a group of objects jointly and learning their accumulated group movement patterns. The recognition of the most delegate [1]movement patterns concerning each group of objects, which are furthermore subjugated to condense the location information. The two-phase and 2D algorithm was proposed to reduce the quantity of distributed data, manipulating the group movement patterns derived to constrict the location sequences of moving objects. The Merge algorithm was proposed to condense the location sequences of a group of moving objects and the Replace algorithm was projected to diminish the entropy of the amalgamated sequence achieved. Discovering the group movement patterns is more difficult than finding the patterns of a single object or all objects, because we need to jointly identify a group of objects and discover their aggregated group movement patterns. The constrained resource of WSNs should also be considered in approaching the moving object clustering problem. In order to study the movement behaviour of dynamic objects, it is important to take a closer look at movement itself. In other words, it is necessary to know what exactly the variables are that define movement, what constraints and external factors affect movement and most importantly to understand what types of movement patterns can be composed from these primitives of movement. Generally, movement patterns include any recognizable spatial and temporal regularity or any interesting relationship in a set of movement data, whereas the proper definition (i.e. the instantiation) of "pattern interestingness" depends on the application domain. Clustering is the unsupervised classification of patterns (observations, data items, or feature vectors) into groups (clusters). The clustering problem has been addressed in many contexts and by researchers in many disciplines; this reflects its broad appeal and usefulness as one of the steps in exploratory data analysis. However, clustering is a difficult problem combinatorial, and differences in assumptions and contexts in different communities has made the transfer of useful generic concepts and methodologies slow to occur Process of reducing the amount of data needed for storage or transmission of a given piece of information typically by use of encoding techniques. Data compression is characterized as either lossy or lossless depending [2]on whether some data is discarded or not, respectively. Data compression can reduce the storage and energy consumption for resource-constrained applications. In Distributed source coding uses joint entropy to encode two nodes' data individually without sharing any data between them; however, it requires prior knowledge of cross correlations of sources.

Our contributions are threefold:

. Different from previous works, we formulate a moving object clustering problem that jointly identifies a group of objects and discovers their movement patterns. The

application-level semantics are useful for various applications, such as data storage and transmission, task scheduling, and network construction.

. To approach the moving object clustering problem, we propose an efficient distributed mining algorithm to minimize the number of groups such that members in each of the discovered groups are highly related by their movement patterns.

. We propose a novel compression algorithm to compress the location data of a group of moving objects with or without loss of information. We formulate the HIR problem to minimize the entropy of location data and explore the Shannon's theorem to solve the HIR problem. We also prove that the proposed compression algorithm obtains the optimal solution of the HIR problem efficiently.

II. EXISTING WORK

Discovering the group movement patterns is more difficult than finding the patterns of a single object or all objects, because we need to jointly identify a group of objects and discover their aggregated group movement patterns. The constrained resource of WSNs should also be considered in approaching the moving object clustering problem. However, few of existing approaches consider these issues simultaneously. On the one hand, the temporal-and-spatial correlations in the movements of moving objects are modeled as sequential patterns in data mining to discover the frequent movement patterns .

On the other hand, previous works, such as measure the similarity among these entire trajectory sequences to group moving objects. Since objects may be close together in some types of terrain, such as gorges, and widely distributed in less rugged areas, their group relationships are distinct in some areas and vague in others. Thus, approaches that perform clustering among entire trajectories may not be able to identify the local group relationships. In addition, most of the above works are centralized algorithms which need to collect all data to a server before processing. Thus, unnecessary and redundant data may be delivered, leading to much more power consumption because data transmission needs more power than data processing in Wireless Sensor Networks (WSNs).

III. PROPOSED APPROACH

We have proposed a clustering algorithm to find the group relationships for query and data aggregation efficiency. The differences of and this work are as follows: First, since the clustering algorithm itself is a centralized algorithm, in this work, we further consider systematically combining multiple local clustering results [4] into a consensus to improve the clustering quality and for use in the update-based tracking network. Second, when a delay is tolerant in the tracking application, a new data management approach is required to offer transmission
efficiency, which also motivates this study. We thus define the problem of compressing the location data of a group of moving objects as the group data compression problem. We first introduce our distributed mining algorithm to approach the moving object clustering problem and discover group movement patterns. Then, based on the discovered group movement patterns, we propose a novel compression algorithm to tackle the group data compression problem.

IV. RELATED WORK

A.MOVEMENT PATTERN MINING

In object tracking applications, many natural phenomenon show that moving objects often exhibit some degree of regularity in their movements. For example, the famous annual wildebeest migration demonstrates that the movement of creatures is temporally and spatially correlated. In addition, biologists have found that many creatures, such as elephants, zebra, whales, and birds, form large social groups when migrating to find food, or for breeding, wintering, or other unknown reasons. These characteristics indicate that the trajectory data of multiple objects may be correlated. Moreover, some research domains, such as the study of animals' social behaviour and wildlife migration are more concerned with a group of animals' movement patterns than each individual's. This raises [3] a new challenge of finding moving animals belonging to the same group and identifying their aggregated movement patterns. Many researchers model the temporal-and-spatial correlations of moving objects as sequential patterns in data mining, and various algorithms have been proposed to discover frequent movement patterns However, such works only consider the movement characteristics of a single object or all objects. Other works, such as take the Euclidean distance to measure the similarity of two entire trajectories, and then derive groups of mobile users based on their movement data.

B.CLUSTERING

Clustering deals with the process of finding possible different groups in a given set, based on similarities or differences among their objects. There is a wide variety of techniques to do clustering. Results are not unique, and they always depend on the purpose of the clustering. The same data can be clustered with different acceptable solutions. Hierarchical clustering, for example, gives several solutions depending on the tree level chosen for the final solution. This clustering method also groups objects according to their connectivity. It uses a pair wise clustering cost function with a dissimilarity measure that emphasizes connectedness in feature space to deal with cluster compactness. This simple approach gives good results with compact clusters.
Data clustering algorithms can be hierarchical. Hierarchical algorithms find successive clusters using previously established clusters. Hierarchical algorithms can be agglomerative ("bottom-up") or divisive ("top-down").

Agglomerative algorithms begin with each element as a separate cluster and merge them into successively larger clusters. Divisive algorithms begin with the whole set and proceed to divide it into successively smaller clusters. Partitioned algorithms typically determine all clusters at once, but can also be used as divisive algorithms in the hierarchical clustering.

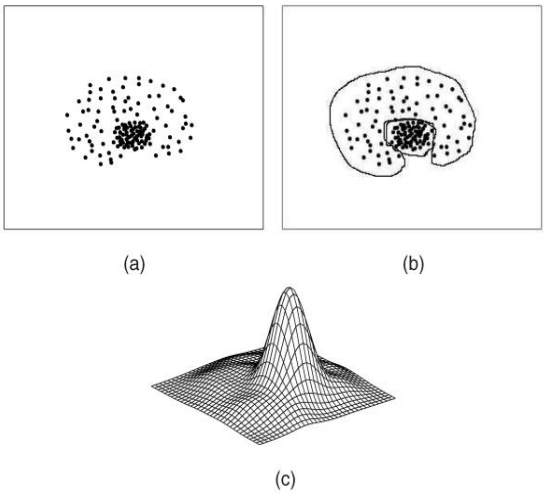

Fig. 1. An example of a data set difficult to cluster using density-based clustering algorithms like Mean Shift. (a) The original data set. (b) The possible clustering solution. (c) Density function.

V. MINING OF GROUP MOVEMENT PATTERNS

we present a new approach to derive groupings of mobile users based on their movement data. We assume that the user movement data are collected by logging location data emitted from mobile devices tracking users. We formally define group pattern as a group of users that are within a distance threshold from one another for at least a minimum duration. To mine group patterns, we first propose two algorithms, namely AGP and VG-growth. In our first set of experiments, it is shown when both the number of users and logging duration are large, AGP and VG-growth are inefficient for the mining group patterns of size two. We therefore propose a framework that summarizes user movement data before group pattern mining. In the second series of experiments, we show that the methods using location summarization reduce the mining overheads for group patterns of size two significantly. We conclude that the cuboid based summarization methods give better performance when the summarized database size is small compared to the original movement database. In addition, we also evaluate the impact of parameters on the mining overhead.

A. The Group Movement Pattern Mining

(GMPMine) Algorithm For mining[5] the group relationship, we propose the GMPMine algorithm shown in Fig. 2 to group objects based on the similarity of their moving patterns. Our algorithm first builds a PST for each object and then leverages the PSTs in grouping objects. We define the similarity score simp of two objects as follows, where S_ denotes the union of significant patterns on two tree, and Lmax is the maximal memory length of PST. simp is the combination of condition probability and Euclidean distance of each significant pattern s ∈ S_. With the definition of simp, we compute the similarity score for each pair of objects and construct an unweighted undirected similarity graph[11] G. A node in G corresponds to an object. An edge between two objects represents that simp of the two objects is higher than the threshold[4] simmin. In addition, we leverage the time-overlapping pruning technique and the noise pruning technique to assist in evaluating the group relationship. The lower time-overlapping ratio between two objects implies the lower probability that the two objects belong to a group. The noise pruning technique utilizes the similarity score between a PST and the PST of certain patterns such as random walk to filter objects. Then we leverages the HCS algorithm together with a min-cut algorithm to partition[1] the similarity graph into highly connected subgraphs, each of which is corresponding to a group. We thereby extract the group relationship among objects. For the details of the algorithm, interested readers can refer to [14].

B. The Cluster Ensembling (CE) Algorithm

In this section, we propose the Cluster Ensembling algorithm to combine multiple local grouping results from the CHs to improve the grouping quality. Our algorithm considers the case that the trajectories of moving objects span multiple sensor clusters and utilizes the Jaccard similarity coefficient to take only relative CHs'[5] opinions in measuring the similarity between pairs of objects. Jaccard Similarity Coefficient is a statistic that is commonly used in information retrieval

GMPMine
Input:Input: Input: Input: S
ˆ ={ Si|0≤i<N }, simmin, Ttarget, simnoise, tomin
Output: Output: Output: G, m
0. G = Ø
1. m = 0
2. /*building a PST for each object and pruning noise*/
3. forfor for each for Si in S
4. Ti = PST_Build(Si)
0. ifififif simp(Ti, Ttarget)>simnoise thenthen then then
5. delete Ti
6. /*constructing a similarity graph on PSTs*/
7. for for for 0 for ≤i< N-1
8. for forfor for i+1≤j< N
9. if if if sim if p(Ti, Tj) >simmin and and and and overlap_time(i,j) > tomin t t then thenhen hen
10. add_edge(i,j) to Graph(V, E)
11. /*extracting highly connected subgraph*/
12. (G,m) = HCS(Graph(V, E)) // G={gi|0≤i<m }
13. /* selecting a group PST GTi for each group gi */
14. for for for 0 for ≤i<m
15. S'= {Sj | oj ∈gi , 0≤j<N }
16. T'= {Tj | oj ∈gi , 0≤j<N }
17. maxarg ' '
18. return G = = = {gi |0≤i<m }, GT = {GTi|0≤i<m }

Fig. 2. The GMPMine Algorithm

for comparing the similarity between two binary vectors [20][13][14]. Jaccard is especially suitable for the applications in which the importance of negative and possible[6] values of a feature is asymmetric, and considering the negative value in both objects has no meaningful contribution to the similarity measurement. After computing the similarity between each pair of objects, we partition the objects and leverage Normal Mutual Information (NMI) to optimize the ensembling result. Note that NMI is an entropy-based measurement criterion between two distributions and is broadly used in the field of information theory. A low NMI value indicates two distributions have only a random association, whereas a higher value indicates they are mutually informative.

Algorithm: Cluster Ensembling
Input: O = {o0,o1,...oN},C = {Gi |0≤i<k }, D = {δi |0≤i<d}
Output : Gδ '
0. init sum[]
1. init SM[][]
2. idx = 0
3. max = 0
4. /*building similarity matrix by Jaccard*/
5. for 0≤i<N-1
6. for i+1≤j<N
7. SM[i,j] = getSij(C)
8. /*select the partition with max∑NMI(Gδ, Gi)*/
9. for 0≤i<d
10. Graph(V, E) = Convert2Graph(SM,δi)
11. i
Gδ = HCS(Graph(V, E))
12. sum[i] =
13. if sum[i] >max then
14. max = sum[i]
15. idx = i
16. Gδ ' =
17. return Gδ ' ∑<≤ kj j i MI GGN 0),(δ idx Gδ
Fig. 3. The Cluster Ensembling Algorithm.

VI COMPRESSION ALGORITHM WITH GROUP MOVEMENT PATTERNS

Transmission of data is one of the most energy expensive tasks in WSNs, data compression is utilized to reduce the amount of delivered data. Therefore, to reduce the amount of delivered data, we propose the 2P2D algorithm. The algorithm includes the sequence merge phase and the entropy reduction phase to compress location sequences vertically and horizontally. In the sequence merge phase, we propose the Merge algorithm to compress the location sequences of a group of moving objects. Since objects with similar movement patterns are identified as a group, their location sequences are similar. The Merge algorithm avoids redundant sending of their locations, and thus, reduces the overall sequence length. It combines the sequences of a group of moving objects by Design of the two-phase and 2D[6] compression algorithm. multiple identical symbols at the same time interval into a single symbol or choosing a qualified symbol to represent them when a tolerance of loss of accuracy is specified by the application. Therefore, the algorithm trims and prunes more items when the group size is larger and the group relationships are more distinct. Besides, in the case that only the location centre of a group of objects is of interest, our approach can find the aggregated value in the phase, instead of transmitting all location sequences back to the sink for post processing[7]. In the entropy reduction phase, we propose the Replace algorithm that utilizes the group movement patterns as the prediction model to further compress the merged sequence.

A. Entropy Reduction Phase

In the entropy reduction phase, we propose the Replace algorithm to minimize the entropy of the merged sequence obtained in the sequence merge phase.

Definition (HIR problem):
Given a sequence S = {si|si € Σ, 0 ≤i < L} and a taglst, an intermediate sequence is a generation of S, denoted by S = {si | 0 ≤ i < L}, where si is equal to si if taglst[i]=0. Otherwise, si' is equal to si or '.'. we derive the first replacement rule—the accumulation rule: Replace all items of symbol σ in where n σ) = nhit σ)

Three Derived Replacement Rules:
1)The HIR problem is to find the intermediate sequence S such that the entropy of S is minimal for all possible intermediate sequences.

2)we derive the second replacement rule—the concentration rule: Replace all predictable items of symbol σ in , where n σ)≤ n '.') or nhit σ) > n σ)=n '.').

3)we derive the third replacement rule—the multiple symbol rule: Replace all of the predictable items of every symbol in

if gain( ) > 0.
To solve the HIR problem, we explore properties of Shannon's entropy to derive three replacement rules that our Replace algorithm [8] leverages to obtain the optimal solution.An important step in any clustering is to select a distance measure, which will determine how the similarity of two elements is calculated. This will influence the shape of the clusters, as some elements may be close to one another according to one distance and further away according to another. For example, in a 2-dimensional space, the distance between the point (x=1, y=0) and the origin (x=0, y=0) is always 1 according to the usual norms, but the distance between the point (x=1, y=1) and the origin can be 2, $\sqrt{2}$ or 1 if you take respectively the 1-norm, 2-norm or infinity-norm distance.

B. The Replace Algorithm:

Based on the observations described in the previous section, we propose the Replace algorithm that leverages the three replacement rules to obtain the optimal solution for the HIR problem.

VII. EXPERIMENTS

We implemented an event driven simulator in C++ with SIM [22] to evaluate our design. In this simulation, we used the Location-Dependent Parameterization of a Random Direction Mobility Model to simulate the roaming behaviour of a group leader and utilized the distance (d) to control the moving range that is the linear distance between the starting point and the end point for the leader. The other members are followers that are uniformly distributed within a specified GDR of the leader. Besides, we inputted objects of random walk in the experiments for making our simulation near to the practical scenarios. In the following experiments, there are 5 groups, each of which contains 5 objects as the grouped, patterned objects and 25 random-walk objects walking in the OTSN. The NMI between the ensembling result and the pre known group relationship of the input workload is used as the evaluation metric. We made the experiments involving the effectiveness of our algorithm and the impacts of PST parameter Lmax,moving distance (d), and GDAR on the grouping quality to test our approach.In the first experiment, we studied the effectiveness of our approach with and without the pruning techniques.

V. CONCLUSIONS

In this work, we exploit the characteristics of group movements to discover the information about groups of moving objects in tracking applications. We propose a distributed mining algorithm, which consists of a local GMPMine algorithm and a CE algorithm, to discover group movement patterns. With the discovered information, we devise the 2P2D algorithm, which comprises a sequence merge phase and an [8] entropy reduction phase. In the sequence merge phase, we propose the Merge algorithm to merge the location sequences of a group of moving objects with the goal of reducing the overall sequence length. In the entropy reduction phase, we formulate the HIR problem and propose a Replace algorithm to tackle the HIR problem. In addition, we devise and prove three replacement rules, with which the Replace algorithm obtains the optimal solution of HIR efficiently. Our experimental results show that the proposed compression algorithm effectively reduces the amount of delivered data and enhances compressibility and, by extension, reduces the energy consumption expense for data transmission in WSNs.

References

[1] S.S. Pradhan, J. Kusuma, and K. Ramchandran, "Distributed Compression in a Dense Microsensor Network," IEEE Signal Processing Magazine, vol. 19, no. 2, pp. 51-60, Mar. 2002.

[2] A. Scaglione and S.D. Servetto, "On the Interdependence of Routing and Data Compression in Multi-Hop Sensor Networks," Proc. Eighth Ann. Int'l Conf. Mobile Computing and Networking, pp. 140-147, 2002.

[3]N. Meratnia and R.A. de By, "A New Perspective on Trajectory Compression Techniques," Proc. ISPRS Commission II and IV, WG II/5, II/6, IV/1 and IV/2 Joint Workshop Spatial, Temporal and Multi- Dimensional Data Modelling and Analysis, Oct. 2003.

[4] S. Baek, G. de Veciana, and X. Su, "Minimizing Energy Consumption in Large-Scale Sensor Networks through Distributed Data Compression and Hierarchical Aggregation," IEEE J. Selected Areas in Comm., vol. 22, no. 6, pp. 1130-1140, Aug. 2004.

[5]C.M. Sadler and M. Martonosi, "Data Compression Algorithms for Energy-Constrained Devices in Delay Tolerant Networks," Proc. ACM Conf. Embedded Networked Sensor Systems, Nov. 2006.

[6]Y. Xu and W.-C. Lee, "Compressing Moving Object Trajectory in Wireless Sensor Networks," Int'l J. Distributed Sensor Networks, vol. 3, no. 2, pp. 151-174, Apr. 2007.

[7] S. Ma, S. Tang, D. Yang, T. Wang, and J. Han, "Combining clustering with moving sequential pattern mining: A novel and efficient technique," 8th PAKDD, pp. 419–423, 2004.

[8] S.-Y. Hwang, Y.-H. Liu, J.-K. Chiu, and E.-P. Lim, "Mining mobile group patterns: A trajectory-based approach," 9th PAKDD, 2005.

[9] V. S. Tseng and K. W. Lin, "Mining temporal moving patterns in object tracking sensor networks," Int. Workshop on Ubiquitous Data Manag., 2005.

[10] D. Ron, Y. Singer, and N. Tishby, "Learning probabilistic automata with variable memory length," 7th annual Conf. on Computational learning theory, Jul. 1994.

[11] G. Bejerano and G. Yona, "Variations on probabilistic suffix trees: statistical modeling and the prediction of protein families," Bioinformatics, vol. 17, no. 1, pp. 23–43, 2001.

[12] S. Dubnov, G. Assayag, O. Lartillot, and G. Bejerano, "Using machinelearning methods for musical style modeling," IEEE Computer Magazine, vol. 36, no. 10, pp. 73–80, Oct. 2003.

[13] P. S. S. Chawla and B. Arunsalam, "Mining for outliers in sequential databases," 6th SIAM Int. Conf. onData Mining (SDM06), Apr. 2006.

[14] G. Mazeroff, V. D. Cerqueira, J. Gregor, and M. Thomason, "Probabilistic trees and automata for application behavior modeling," 41st ACM Southeast Regional Conf. Proceedings, 2003.