ISSN:2231-0711
www.ijcset.net

Analysis and Grouping of Movable Object Patterns Using Similarity Measure of Trajectories in Wireless Sensor Networks

Vijayasaradhi Thommandru[1], Subramanyam Kodukula[1], Vamsidhar Talasila *[2] and Anusha Peruri *[2]
[1], *Dept of CSE, K L University.*
Green Fields, Vijayawada, India

*[2]*M.Tech (CSE)*
Green Fields, K.L.Univerisity,
Vijayawada, India.

Abstract — **Many natural phenomena show that objects often exhibit some degree of regularity in their movements. However, previous works focus on finding the movement patterns of each single object or all objects. In this paper, we propose an efficient distributed mining algorithm to jointly identify a group of moving objects and discover their movement patterns in wireless sensor networks. As the movements of an object are regular, the object's next location can be predicted based on its preceding locations. We model the regularity by using the Variable Length Hidden Markov Model (VLHMM). If a pattern occurs more frequently, it carries more information about the movements of an object. To find out these information patterns, we first define the pattern as a significant pattern by adapting a Prediction Suffix Tree (PST) with improved smoothening factor. Our distributed mining algorithm comprises of Group Trajectory Mine (GTM) and Cluster Ensemble (CE) algorithms. In GTM, we propose a new similarity measure known as minkowski, which is used to compute the similarity of moving objects. In CE phase, our algorithm combines the multiple local grouping results which come from the GTM algorithm. We further leverage the mining results to track moving objects efficiently.**

Keywords **--- Distributed Clustering, Similarity measure, Object Tracking, WSN, Prediction Suffix Tree**

1. INTRODUCTION

Recent advances in wireless sensors and mobile technologies have to equip their devices with positioning sensors that utilize the global positioning system (GPS). Accurate positioning of mobile devices makes the way for the deployment of location based services and applications like object tracking, environmental monitoring etc. These applications flood in with large amounts of moving object data. This results in transmission and storage challenges in WSN's, Since it has constrained resources such as on-board, non - refreshable battery power, computation capability and storage.

Object tracking is an important application of WSN's. Tracking consists of deleting and monitoring locations of real world objects, possibly using several types of sensing such as acoustic, seismic, electromagnetic etc. Object tracking sensor networks have two critical operations. First, monitoring: sensor nodes are required to detect and track the movement states of mobile objects. Second, reporting : the nodes that sense the objects need to report their discoveries to the applications. These two operations are interleaved during the entire object tracking process.

In object tracking applications many natural phenomena shows that objects often exhibit some degree of regularity in their movements. The movements of creatures are temporally and spatially correlated. 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 or wintering. These characteristics indicate that the trajectory data of multiple objects may be correlated for biological applications. More over some research domains, such as the study of animals behavior and wild life migration[1], [2], are more concerned with the movement patterns of group of animals, not individuals; hence, tracking each object is unnecessary in this case. This raises a new challenge of finding moving animals belonging to the same group and identifying their aggregated group movement patterns. Therefore, under the assumption that objects with similar movement patterns are regarded as a group, we define the moving object clustering problem as given the movement trajectories of objects, partitioning the objects into non overlapped groups such that the number of groups is minimized. Since there are inherent variations in the number of groups and their sizes (e.g., elephant herds may contain 8-100 individuals, depending on the environment and family size [11]), it is difficult to predetermine these two parameters. Therefore, we use the HCS algorithm [12] to cluster objects efficiently without pre specifying the number of groups or their sizes.

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. 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 [3], [4], [5], [6]. On the other hand, previous works, such as [7], [8], [9], measure the similarity among these entire trajectory sequences to group moving objects.

In this paper, we first introduce our distributed mining algorithm to approach the moving object clustering problem and discover group movement patterns. Previous works focus on finding the significant patterns by using the prediction suffix trees (PSTs). But it has a smoothening problem. To overcome this problem, we are using improved smoothening factor in constructing PSTs. Our distributed mining algorithm comprises a Group Trajectory Mine (GTM) and a Cluster Ensemble (CE) algorithms. The GTM algorithm discovers the local group movement patterns by using a novel similarity measure. In previous paper, to measure the distance between significant patterns,

Euclidean distance is used. But this is not appropriate when objects have high-dimensions. So, we propose a similarity measure by using the minkowski distance. CE algorithm combines the local grouping results to remove inconsistency and improve the grouping quality by using the Jaccard coefficient. In contrast to approaches that perform clustering on entire trajectories at a central server, the proposed algorithm discovers the local group relationship in a distributed manner on sensor nodes

The contribution of this paper is threefold. First, we propose a prediction suffix tree with improved smoothening factor to find out the significant movement pattern. Second, we propose a new pair wise measure distance based on pattern similarity. To compute the similarity of moving objects we use minkowski distance. Third, we use the discovered information to track moving objects efficiently. The remainder of this paper is organized as follows: In Section 2, we review related works. Section 3, provide an overview of our network model and location model. In Section 4, we describe the design of our Group trajectory mining algorithm. Section 5, we summarize our conclusions.

2. RELATED WORK

2.1 *Movement Pattern Mining*

The temporal-and-spatial correlations and the regularity in the trajectory data sets of moving objects are often modeled as sequential patterns for use in data mining. Agrawal and Srikant [13] first defined the sequential pattern mining problem and proposed an Apriori-like algorithm to mine frequent sequential patterns. Han et al. proposed Free Span [14], which is an FP-growth-based algorithm that addresses the sequential pattern mining problem by considering the pattern-projection method. For handling the uncertainty in trajectories of mobile objects, Yang and Hu [15] developed a new match measure and proposed Trajpattern to mine sequential patterns from imprecise trajectories. Moreover, a number of research works have been elaborated upon mining traversal patterns for various applications. For example, Chen et al. [16] proposed the FS and SS algorithms for mining path traversal patterns in a Web environment while Peng and Chen [17] proposed an incremental algorithm to mine user moving patterns for data allocation in a mobile computing system. However, sequential patterns or path traversal patterns do not provide sufficient information for location prediction or clustering. To discover significant patterns for location prediction, Morzy proposed Apriori-Traj [18] and Traj-Prefix Span [19] to mine frequent trajectories, where consecutive items of a frequent trajectory are also adjacent in the original trajectory data. Meanwhile, the approach in [20] extracts T-patterns from spatial temporal data sets to provide concise descriptions of frequent movements. Tseng and Lin [21] proposed the TMP-Mine algorithm for discovering the temporal movement patterns of objects.

2.2 *Trajectory Clustering*

Recently, clustering based on objects' movement behavior has attracted more attention. For example, Li et al. [22] employ Moving Micro clusters (MMC) to discover and maintain a cluster of moving objects online. Meanwhile, Lee et al. [23] proposed trajectory clustering to discover popular movement paths. Clustering similar trajectory sequences to discover group relationships is closely related to our problem. Wang et al. [24] transform the location sequences into a transaction-like data on users and based on which to obtain a valid group. However, the proposed AGP and VG-growth algorithms are Apriori-like or FP-growth based algorithms that suffer from high computing cost and memory demand. Nanni and Pedreschi [25] apply a density-based clustering algorithm to the trajectory clustering problem based on the average euclidean distance of two trajectories. However, the above works that discover group information based on the proportion of the time a group of users stay close together or the average euclidean distance of the entire trajectories may not reveal the local group relationships, which are required for many applications.

2.3 *Similarity Measure*

Identifying the similarity (or distance) between two trajectories is essential for clustering. Computing the average euclidean distance of two geometric trajectories is a simple and useful approach. Nevertheless, the geometric coordinates are expensive and not always available. Other approaches, such as EDR, LCSS, and DTW, are widely used to compute the similarity of symbolic sequences [26]. However, the above dynamic programming approaches suffer from scalability problem [27]. Therefore, approximation or summarization techniques are used to represent original data for providing scalability. However, when projecting each data sequence into a vector space of sequential patterns, the importance of a sequential pattern regarding to each data sequence is not mentioned. In addition, Yang and Wang [28] employ a probabilistic suffix tree to learn the structural features of sequences and proposed a new similarity measure which computes the similarity of a probabilistic suffix tree and a sequence. Their clustering algorithm iteratively identifies a sequence to a cluster and adjusts the representative probabilistic suffix tree for each cluster. However, the generated clusters may overlap which differentiates their objective from ours.

2.4 *Distributed Clustering*

Distributed clustering is an important research topic. Most of the approaches proposed in the literature focus on seeking a combination of multiple clustering results to achieve better clustering quality, stability, and scalability. For example, Strehl and Ghosh [29] introduced and formulated the clustering ensemble problem to a hypergraph partitioning problem, and proposed CSPA, HGPA, and MCLA to compute the best K-partition of the graph. Ayad et al. [30] presented a probabilistic model to combine cluster ensembles by utilizing information theoretic measures. Fred and Jain [31] combine multiple runs of the K-means algorithm with random initializations and random numbers to obtain the final consensus partition. Fern and Brodley [32] apply random projection to the high dimensional data and cluster the reduced data by using EM for a single run of clustering. The Collective PCA technique [33] proposed by Kargupta et al. is applied to reduce the vector dimension for distributed clustering of high dimensional heterogeneous data. The data types of the above works [30], [31], [32], [33], [34] are most integer vector or categorical data, and their related issues are thereby different from ours. In addition, previous works that require a predetermined k in their clustering or Ensemble algorithms are not suitable for our applications. Besides, although the local grouping results in a vector of

integers, each of which represents the mapping between an object and its belonging group, dimension reduction like Collective PCA [33] is unnecessary in our case.

3. PRELIMINARIES

3.1 *Hierarchical Object Tracking Sensor Network*

Many researchers believe that a hierarchical architecture provides better coverage and scalability, and also extends the network lifetime of WSNs [35], [36]. In a hierarchical WSN, such as that proposed in [37], the energy, computing, and storage capacity of sensors are heterogeneous. A high end sophisticated node, such as Intel Stargate [38], is assigned as a CH to perform high complexity tasks; while a resource constrained node, such as Mica2 mote [39], performs the sensing and low complexity tasks. In this work, we adopt a hierarchical and cluster-based network structure with K layers. As shown in Fig. 1a, the nodes are clustered in each level, and each cluster is a mesh network of fixed size, i.e., each cluster is a set of n x n sensors. We assume that each sensor in a cluster has a locally unique ID, and denote the sensor IDs by an alphabet Σ . Fig. 1b shows an example of a two-layer network structure, where each cluster contains 16 nodes whose IDs are identified by Σ = { a, b,....., p }.

The sensor nodes at each level are divided into some equal number of clusters and we assume that each cluster in that level having a individual cluster heads called Virtual Center Heads (VCHs). VCHs acts like a bridge between sensor nodes and Cluster Head (CH). All the VCHs at that level together form a global cluster head. These VCHs will collect the location data from the respective clusters. VCHs are also used to combined and transmit the location data of the object to its Global Cluster Head effectively. From the Global Cluster Head the data will be transmit to the Cluster head of the next level. Here we are considering the transmission traffic as a effective parameter and reduce it by using these VCHs. First the object location data will be collected by the sensors and here for 16 sensor nodes , 4 VCHs are used at each level to transmit the location data to its Cluster Head independently. By introducing this concept we will able to reduce the so much of network traffic in terms of packets and reduce the work load at the Cluster Head (CH). Whenever the work load of Cluster Head was reduced then automatically the data acquisition speed at the sink also increased. By using these VCHs overlapping overhead of location data also minimized at the time of transmission.

In this work, an object is defined as a target, such as an animal or a bird, that is recognizable and trackable by the tracking network. A cluster of sensors communicate with each other by using multi hop routing, and wake up on their duty cycles to carry out a given task [40]. They collaboratively gather or relay remote information to a base station called a sink. Take the tracking application for example. When a sensor wakes up and detects an object of interest, it transmits the location data of the object to its CH and then enters the sleep mode. The CH aggregates the data and forwards it to the CH of the upper layer. The process is repeated until the sink node receives the location data. The data flow is as shown in Fig. 1 [43]. When a task of discovering the group relationships of objects is assigned, it is unnecessary to transmit all the location data to the sink for post processing. In our design, CHs collect the location data for a period and generate location sequence data sets locally. Then, based on the data sets, our mining algorithm tries to discover the group relationships about the objects of interest. Geometric models and symbolic models are widely used to represent the location of objects [41]. A geometric location denotes precise 2-dimension or 3-dimension coordinates; while a symbolic location represents an area, such as the sensing area of a sensor or a group of sensors, defined by the applications. Since the accurate geometric location is not easy to obtain, in this work, we employ a symbolic model and take the sensors' IDs as the locations of an object of interest. Sensors are closely deployed to ensure complete coverage of the monitored area, but this causes consistency and redundancy problems. Techniques like the Received Signal Strength (RSS) [42] simply estimate an object's location based on the ID of the sensor with the strongest signal and eliminate unnecessary transmissions. The trajectory of a moving object is thus model as an ordered sequence of sensor IDs, i.e., a location sequence denoted by s = $\sigma_0,\sigma_1....\sigma_{l-1}$, where $\boldsymbol{\sigma_i} \in \boldsymbol{\Sigma}$ and l is the sequence length.

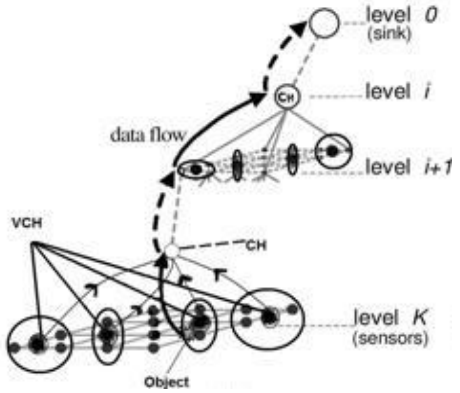

Fig. 1. The hierarchical and cluster-based network structure and the data flow of an update-based tracking network.

3.2 *Variable Length Hidden Markov Model (VLHMM) and Prediction Suffix Tree (PST)*

If the movements of an object are regular, the object's next location can be predicted based on its preceding locations. We model the regularity by using the Variable Length Hidden Markov Model (VLHMM). The VLHMM is superior in its efficiency and accuracy of modeling multivariate time-series data with highly-varied dynamics. Our motivation of developing VLHMM comes from the following observations: (1) although the first-order HMM (HMM) is efficient in learning, it is inaccurate in modeling; (2) the fixed-length high-order HMM (n-HMM) is accurate in modeling but, because of its huge number of parameters, is not efficient; (3) Various variants to n-HMM, including the Mixture Transition Distribution (MTD), introduces mathematical constraints to simplify n-HMM. Although the simplification reduces model complexity, it also reduces the generality and accuracy of modeling. Unlike these variants that compromise between HMM and n-HMM, VLHMM is both accurate and efficient.

Variable Length Markov Model (VLMM) is an "observable" Markov model and can only model sequences of discrete values; whereas VLHMM is a "hidden" model and can model sequences of discrete/continuous and

scalar/vector values. Learning VLMM only needs to optimize the Minimum-Entropy criterion, whereas learning VLHMM needs to optimize both the Minimum-Entropy and the Maximum-Likelihood criteria. In some papers, VLMM was used to learn a sequence of continuous/vector values. This is done by (1) discarding the order of the values in the sequence, (2) clustering the values into a finite number of clusters, (3) substituting each value of the sequence by its cluster index, and (4) learning *VLMM* from the sequence of cluster indices. However, this method is far from optimal, because the error introduced in the clustering step (usually called quantization or discretization) caused by discarding the order information may be very large and is out of control. In fact, learning a hidden Markovian model (HMM, n-HMM, or VLHMM) is itself a clustering process, where each cluster is represented by an output probability distribution function. Due to these advantages, a wider variety of patterns can be mined from VLHMM than from other models.

When a pattern "**s"** occurs more frequently, it carries more information about the movements of the object and is thus more desirable for the purpose of prediction. To find the informative patterns, we first define a pattern as a significant movement pattern if its occurrence probability is above a minimal threshold. To learn the significant movement patterns, we adapt Prediction Suffix Tree (PST). Previous works focus on finding the significant patterns by using the probability suffix trees(PSTs). PST provides very simple solution to the problem, probability smoothening. It has some drawbacks. First, the same constant value γ_{min} is added to every probability, whatever the observed frequency and the probability of the event are. Second, the same floor probability γ_{min} is assigned to all unseen events, whatever the suffix is. To overcome these drawbacks, we propose PST with improved smoothening factor known as Kneser-Nay-back-off smoothing. Kneser and Ney showed that using the back-off distribution even if the main distribution is not null leads to a better model. We then have:

$$P(\sigma|s) = \begin{cases} \frac{c(s,\sigma)-d_C}{\sum_{\sigma\epsilon\Sigma} c(s,\sigma)} + \alpha^1(s)\beta(s,\sigma) & if\ c(s,\sigma) > 0 \\ \alpha^1(s)\beta(s,\sigma) & otherwise \end{cases}$$

Where $\alpha^1(s)$ is a normalization factor.

The PST building algorithm extracts significant patterns from a sequence (or a dataset) and prunes unnecessary nodes during tree construction, and then generates a PST. A PST over Σ is a non-empty tree, in which nodes vary in degree between zero and |Σ|. Each edge in the tree is labeled by a single symbol of Σ such that no symbol can be represented by more than one edge branching out from any single node. Each node in the tree is labeled by a string s, which is a sequence generated by walking up the tree from that node to the root. In addition, each node also contains the conditional empirical probabilities, i.e., P (σ | s), for every σ ε Σ. Note that the maximal memory length (or maximum height) of a PST is specified by L_{max} ; and the criteria for the PST building algorithm to create a node of a sequence s are as follows: 1) The occurrence probability of s must be above the minimal support P_{min} ; 2) the transition behavior P (σ | s) must differ significantly from that of the parent.

Fig. 2 shows the PST_build algorithm. The input of the algorithm includes the PST parameters (P_{min}, α, γ_{min}, r, L_{max}) and a location sequence S; P_{min} is the minimal support of patterns; α is a parameter that together with the smoothing probability defines the significance threshold; γ_{min} is the smoothing probability; r is a measure of the difference between the conditional probability of the candidate and its father node; and L_{max} is the maximal memory length (maximal height of the tree). The output of the PST_build algorithm is a PST T that contains the significant patterns in S. The algorithm starts by initializing T and then computes the candidate patterns with Length 1 (Lines 3-5), where getP(σ,S) is a function that returns P(σ) based on S. As shown in Lines 10-11, the algorithm checks whether each candidate is qualified to be a node in the tree. If the candidate s is qualified, a node associated with s and the nodes on the path to the root node are added to the PST (Lines 12-16). Next, the algorithm extends the candidates, as shown in Lines (18-20), Finally, it smoothens the conditional probabilities of all nodes in T to avoid zero conditional empirical probabilities (Lines 22-24), where $c(s,\sigma)$ is the number of times σ was seen after the suffix s and d_C is the discount parameter, which may depend on $c(s,\sigma)$. $\alpha^1(s)$ is a normalization factor and $\beta(s,\sigma)$ is the back-off distribution. A node labeled with a sequence s is a significant pattern of T, denoted by s ε T.

Algorithm: PST_build
Input: $P_{min,}$ α, γ_{min}, r, $L_{max,}$ Σ, S=s_0,s_1,….s_{n-1}
Output: T

0. /* Initialization */
1. T = node (root, S)
2. X = Ǿ
3. **for each** σ ε Σ
4. **if** get P(σ, S) ≥ P_{min} **then**
5. add σ to X
6. /* Building the PST skeleton */
7. **while** X ≠ Ǿ
8. **for each** s ε X
9. remove s from X
10. **if** exists σ ϵ Σ **and** getCP(σ, s, S) ≥ (1+α)γ_{min} **and**
11. $\frac{getCP(\sigma,s,S^1)}{getCP(\sigma,suf(s),S)} > r \quad \frac{getCP(\sigma,s,S^1)}{getCP(\sigma,suf(s),S)} > 1/r$ **then**
12. add node(s, S)to T
13. S^1=S
14. **while** exists suf (S^1)
15. add node(suf(S^1),S) to T
16. S^1= suf(S^1)
17. **If** |S|<L_{max} **then**
18. **for each** σ ϵ Σ
19. **If** get P(σ s, S)≥ P_{min} **then**
20. add σ s to X
21. /* smoothing the conditional probabilities*/
22. **for each** node nd in T
23. **for each** σ ϵ Σ
24. nd.P(σ|s)=
$$\begin{cases} \frac{c(s,\sigma)-d_C}{\sum_{\sigma\epsilon\Sigma} c(s,\sigma)} + \alpha^1(s)\beta(s,\sigma) & if\ c(s,\sigma) > 0 \\ \alpha^1(s)\beta(s,\sigma) & otherwise \end{cases}$$
25. **return** T

Fig.2. The PST_build algorithm.

PST is useful and efficient in predicting the next item of a sequence. The algorithm's Computational overhead is limited by the height of a PST so that it is suitable for sensor nodes. PST is frequently used in predicting the occurrence probability of a given sequence, which provides us important information in similarity comparison. The occurrence probability of a sequence s regarding to a PST T, denoted by $P^T(s)$, is the prediction of the occurrence probability of s based on T. For example, the occurrence probability P^T ("avsno") is computed as follows:

$$P^T(\text{“avsno”}) = P(a)\,P(v/a)\,P(s/av)\,P(n/avs)\,P(o/avsn)$$

$$= \varphi^{\varepsilon a}\,\varphi^{\varepsilon av}\varphi^{\varepsilon avs}\,\varphi^{\varepsilon avsn}\,\varphi^{\varepsilon avsno}$$

$$= 0.67 \times 0.5 \times 1 \times 1 \times 1 = 0.33$$

4. DISCOVERY OF GROUP MOVEMENT PATTERNS

4.1 *Design of the Distributed Trajectory Mining Algorithm*

In this work, we model the movement of an object by a VLHMM, and use a PST to mine the significant movement patterns. A set of moving objects is regarded as belonging to the same group if they share similar movement patterns. In this section, we first propose a new similarity measure to define the pair wise similarity of moving objects. The advantages of the new proposed similarity measure sim_p include its efficiency and its accuracy. First, sim_p compares the similarity of two objects based on their significant movement patterns instead of their entire location sequences. Thus, sim_p can provide efficiency for the applications with evolving and evolutionary similarity relationships. Second, it considers the importance of each movement pattern regarding to each individual object so that it achieves better accuracy in similarity comparison. With the definition of sim_p, two objects are similar if their similarity score is above a minimal threshold. A set of objects is regarded as a group if each object is similar to at least half the members of the same group.

To tackle the problem of discovering groups of moving objects, we propose a distributed mining algorithm comprised of a GTMine algorithm and a CE algorithm as shown in Fig. 3. The GTMine algorithm uses a PST to generate the significant movement patterns and computes the pair wise similarity of moving objects by using sim_p. It utilizes the HCS algorithm to cluster the moving objects into non overlapped groups. The sink applies the CE algorithm to combine the local grouping results. The CE algorithm utilizes the Jaccard similarity coefficient to measure the similarity between a pair of objects, and normalized mutual information (NMI) to derive the final ensembling result G_δ^1 .

At the local end, CH_i performs the GTMine algorithm to generate local grouping result G_i while the sink performs the Cluster Ensemble algorithm to combine the local grouping results into a consensus final result G_δ^1.

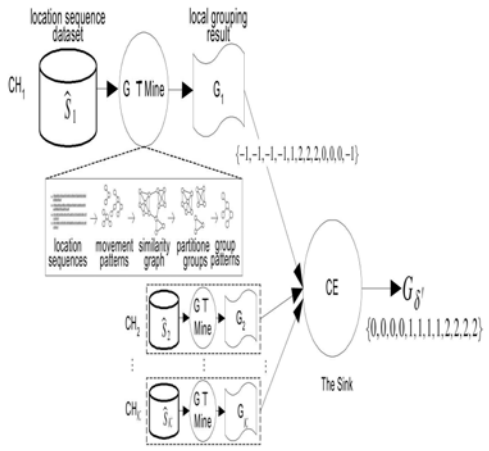

Fig. 3. The framework of our distributed mining algorithm:

4.2 *Similarity Measurement Using Minkowski*

In this work, we use a PST to mine significant movement patterns of an object, where a significant movement pattern is a subsequence with occurrence probability higher than a minimal threshold. Each node of a PST represents a significant movement pattern and carries its conditional probabilities, and all nodes of a PST provide the precise information about the predicted occurrence probability of a given pattern. To provide better discrimination accuracy, we propose a new similarity measure sim_p that adequately and skillfully utilizes the information carried by PSTs to measure the similarity of two objects. The design concepts of sim_p are simple and useful. The importance of a pattern s is modeled by using the predicted occurrence probability, i.e., P^T (S),while the difference of a pattern is defined over all of the dimensions, i.e., P^T (σ | s) . Based on the two concepts, we define the distance a pattern 's' associated with two objects o_i and o_j as

$$\boldsymbol{d(s)} = \sqrt[3]{\sum_{\sigma\epsilon\Sigma}(\boldsymbol{P^{T_i}(s\sigma) - P^{T_j}(s\sigma)})^2} \quad \text{--------} \quad \textbf{(1)}$$

then i.e.,

$$\sqrt[3]{\sum_{\sigma\epsilon\Sigma}(\boldsymbol{P^{T_i}(s)} \times \boldsymbol{P^{T_i}(\sigma|s)} - \boldsymbol{P^{T_j}(s)} \times \boldsymbol{P^{T_j}(\sigma|s)})^2}$$

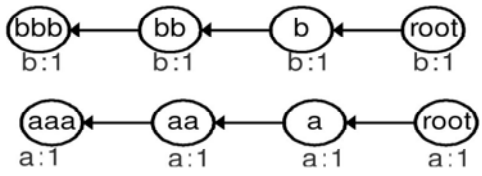

Fig. 4. The maximal value of $\Sigma_{s\epsilon S1}$ d(S) of the two PSTs is $6 + \sqrt{2}$

where T_i and T_j are their respective PSTs. d(s) is the minkowski distance of products of the importance and difference over 'Σ' related to o_i and o_j. Note that since the similarity of two objects is symmetric in our applications,a symmetric measure, such as minkowski distance, is more desirable. Furthermore, for a pattern s ϵ T_i, P^T_i (s) is a

significant value because the occurrence probability of 's' is higher than the minimal support P_{min}. If o_i and o_j share the pattern 's', we have $s \in T_i$ and $s \in T_j$, respectively, such that $P^{Ti}(s)$ and $P^{Tj}(s)$ are non-negligible and meaningful in the similarity comparison. Thus, we define the similarity score of o_i and o_j by using all of their significant movement patterns as follows:

$$Sim_p(O_i,O_j) = - \log \frac{\sum_{s \epsilon S} \sqrt[3]{\sum_{\sigma \epsilon \Sigma} (P^{Ti}(s\sigma) - P^{Tj}(s\sigma))^2}}{2 L_{max} + \sqrt{2}} \quad \text{--- (2)}$$

where S^1 denotes the union of significant movement patterns (nodes) of both objects (trees). We sum d(s) for all $s \in S^1$ as the distance between two PSTs and normalize it by its maximal value, i.e., $2L_{max} + \sqrt{2}$. For example, the PSTs shown in Fig. 4 are built for two sequences "aa...a" and "bb...b" with $\alpha = 0$, $r = 1$, and $L_{max} = 3$. The distance between two PSTs is $6 + \sqrt{2}$, where $S^1 = \{$"a", "aa", "aaa", "b", "bb", "bbb"$\}$. Thereafter, we take the negative log of the distance between two PSTs as the similarity score such that a larger value of the similarity score implies a stronger similar relationship, and vice versa. The worst-case complexity of computing sim_p is $O(L_{max} \text{ X } |\Sigma| \text{ X } |S^1|)$. Therefore, the complexity of building a PST and scoring the similarity between two PSTs is $O(L_{max} \text{ X } |\Sigma| \text{ X } |S^1| + l)$, where l is the sequence length.

4.3 *Group Trajectory Mine (GTM) Algorithm*

We now describe the GTMine algorithm, which identifies groups of objects and determines their movement patterns. The minimal similarity threshold (sim_{min}) is the lower limit of the similarity between two objects belonging to the same group. Let $O = \{o_0,o_1,...,o_{N-1}\}$ denote the objects of interest and $\pi(o_i)$ denote the mapping of the group ID and object o_i. The GTMine algorithm generates the grouping result G and the associated group movement patterns GT. Specifically, G is composed of m disjoint groups of objects over O, denoted by $G = \{g_0,g_1,...,g_{m-1}\}$, where $g_i = \{ o_j | \pi(o_j) = i, o_j \epsilon O\}$. The group movement patterns associated g_i is denoted by GT_i, and $GT = \{GT_0,GT_1,...,GT_{m-1}\}$ denotes the group movement patterns for the m groups. The GTMine algorithm is comprised of four steps. First, we extract the movement patterns of each object from the location sequence. Second, we construct a similarity graph in which similar objects are connected by an edge. Third, we extract highly connected components to derive the group information. Fourth, we construct a group PST for each group in order to conserve the memory space. The above steps extract the group information and object movement patterns. In constructing Group Prediction Suffix tree step, we retain the most representative PST of a group of objects for storage efficiency.

4.4 *The Cluster Ensemble (CE) Algorithm*

In the previous section, each CH collects location data and generates local group results with the proposed GTMine algorithm. Since the objects may pass through only partial sensor clusters and have different movement patterns in different clusters, the local grouping results may be inconsistent. For example, if objects in a sensor cluster walk close with together across a canyon, it is reasonable to consider them a group. In contrast, objects scattered in grassland may not be identified as a group. Furthermore, in the case where a group of objects moves across the margin of a sensor cluster, it is difficult to find their group relationships. Therefore, we propose the CE algorithm to combine multiple local grouping results. The algorithm solves the inconsistency problem and improves the grouping quality. The ensemble problem involves finding the partition of O that contains the most information about the local grouping results. Let C denote the ensemble of the local grouping results, represented as $C = \{G_0,G_1,...,G_{k-1}\}$, where k denotes the ensemble size, i.e., the total number of CHs. we utilize NMI to evaluate the grouping quality. The mutual information of G_i and G_j is a measure of the amount of information shared by the two results

$$MI(G_i,G_j) = \sum_{a=0}^{m_i-1} \sum_{b=0}^{m_j-1} P^1(a,b) \log \frac{P^1(a,b)}{p^1(a) \times p^1(b)},$$

where $p^1(a)$ denotes the probability function of G_i, defined as

$$p^1(a) = \frac{|g_a^i|}{|O|};$$

and $P^1(a,b)$ is the joint probability distribution function of G_i and G_j, defined as

$$P^1(a,b) = \frac{|g_a^i \cap g_b^j|}{|O|}.$$

For a probable ensembling result G, the summation of NMIs of G and every $G_i \epsilon C$ represents the amount of information that G contains with respect to C. Therefore, the ensembling result G^1 that contains the most information about C is given by,

$$G^1 = \arg\max \sum_{i=0}^{k-1} NMI(G\boldsymbol{i}, G),$$

The algorithm leverages the information in C to generate the ensembling result G_δ , and trades off the grouping quality against the computation cost by adjusting the partition parameter D, i.e., a set of thresholds with values in the range [0,1]. A finer grained configuration of D achieves a better grouping quality, but the penalty is a higher computation cost. The proposed CE algorithm is comprised of three steps. First, it utilizes the Jaccard similarity coefficient to measure the similarity of each pair of objects. Second, it partitions the objects for every $\delta \epsilon D$; and third, it employs NMI to optimize the ensembling result. In the following, we describe the three steps in detail. The sink node uses the CE algorithm to combine the local grouping results. It then assigns a global group ID to each group and sends the group information to the CHs for subsequent collection of the location data.

5. CONCLUSIONS

In this work, we exploit the characteristics of group movements to discover the information about groups of moving objects in an Object Tracking Sensor Network (OTSN). In contrast to the centralized mining technique, we mine the group information in a distributed manner. We propose a novel distributed mining algorithm, which consists of a local GTMine algorithm and a CE algorithm, to discover group information. Our algorithm mines object movement patterns as well as group information and the estimated group dispersion radius. Our experimental results show that the proposed mining technique achieves good grouping quality. Prediction Suffix tree and VLHMM

models are very useful to predict the location sequence much accurately. Minkowski distance which is also very effective distance measure for movable objects because of its high dimensionality supported feature.

ACKNOWLEDGEMENTS

We are greatly delighted to place my most profound appreciation to Dr. K. Satyanarayana **Chancellor of K.L.University**, Dr. K. Raja Sekhara Rao **Principal**, S. Venkateswaralu **Head of the department of CSE** and Dr. K. Subramanyam **coordinator for M.Tech** under their guidance and encouragement and kindness in giving us the opportunity to carry out the paper. Their pleasure nature, directions, concerns towards us and their readiness to share ideas enthused us and rejuvenated our efforts towards our goal. We also thank the anonymous references of this paper for their valuable comments.

REFERENCES

[1] G. Shannon, B. Page, K. Duffy, and R. Slotow, "African Elephant Home Range and Habitat Selection in Pongola Game Reserve, South Africa," African Zoology, vol. 41 ,no. 1, pp. 37- 44, Apr. 2006.

[2] C. Roux and R.T.F. Bernard, "Home Range Size, Spatial Distribution and Habitat Use of Elephants in Two Enclosed Game Reserves in the Eastern Cape Province, South Africa," African J. Ecology, vol. 47, no. 2, pp. 146-153, June 2009.

[3]J.Yang and M. Hu, "Trajpattern: Mining Sequential Patterns from Imprecise Trajectories of Mobile Objects," Proc. 10th Int'l Conf. Extending Database Technology, pp. 664-681, Mar.2006.

[4] M. Morzy, "Mining Frequent Trajectories of Moving Objects for Location Prediction," Proc.5th Int'l Conf. Machine Learning and Mining in Pattern Recognition, pp. 667-680, July 2007.

[5] F. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi, "Trajectory Pattern Mining," Proc. ACM SIGKDD, pp. 330-339, 2007.

[6] V.S. Tseng and K.W. Lin, "Energy Efficient Strategies for Object Tracking in sensor networks: A Data Mining pproach," J. Systems and Software, vol. 80, no. 10, pp.1678-1698, 2007.

[7] L. Chen, M. Tamer O ¨ zsu, and V. Oria, "Robust and Fast Similarity Search for Moving Object Trajectories," Proc. ACM SIGMOD, pp. 491-502, 2005.

[8] Y. Wang, E.-P. Lim, and S.-Y. Hwang, "Efficient Mining of Group Patterns from User Movement Data," Data Knowledge Eng., vol. 57, no. 3, pp. 240-282, 2006.

[9] M.Nanni,D.Pedreschi, "Time-Focused Clustering of Trajectories of Moving Objects,"J. Intelligent Info Systems, vol. 27, no. 3, pp. 267-289, 2006.

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

[11] http://www.defenders.org/wildlife_and_habitat/ Wildlife /elephant.php, 2010.

[12]E.Hartuv and R.Shamir, "A Clustering Algorithm Based on the Graph Connectivity," Information Processing Letters, vol. 76, nos. 4-6, pp. 175-181, 2000.

[13] R.Agrawal and R.Srikant, "Mining Sequential Patterns," Proc. 11th Int'l Conf. Data Eng., pp. 3-14, 1995.

[14] J.Han,J.Pei, B.Mortazavi-Asl, Q.Chen, U. Dayal, and M.Hsu, "Freespan: Frequent Pattern-Projected Sequential Pattern Mining," Proc. Sixth ACM SIGKDD, pp. 355-359, 2000.

[15] J.Yang and M.Hu, " Trajpattern: Mining Sequential Patterns from Imprecise Trajectories of Mobile Objects," Proc. 10th Int'l Conf. Extending Database Technology, pp. 664-681, Mar. 2006.

[16] M.-S. Chen, J.S. Park, and P.S. Yu, "Efficient Data Mining for Path Traversal Patterns," Knowledge and Data Eng., vol. 10, no. 2, pp. 209-221, 1998.

[17] W.-C. Peng and M.-S. Chen, "Developing Data Allocation Schemes by Incremental Mining of User Moving Patterns in a Mobile Computing System," IEEE Trans.Knowledge and Data Eng., vol. 15, no. 1, pp. 70-85, Jan./Feb. 2003.

[18] M.Morzy, "Prediction of Moving Object Location Based on Frequent Trajectories,"Proc. 21st Int'l Symp. Computer and Information Sciences, pp. 583-592, Nov. 2006.

[19] M. Morzy, "Mining Frequent Trajectories of Moving Objects for Location Prediction," Proc. Fifth Int'l Conf. Machine Learning and Data Mining in Pattern Recognition, pp. 667- 680, July 2007.

[20] F. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi, "Trajectory Pattern Mining,"Proc.13th ACM SIGKDD, pp. 330-339, 2007.

[21] V.S. Tseng and K.W. Lin, "Energy Efficient Strategies for Object Tracking in Sensor Networks: A Data Mining Approach," J. Systems and Software, vol. 80, no. 10, pp. 1678-1698, 2007.

[22] Y. Li, J. Han, and J. Yang, "Clustering Moving Objects," Proc. 10th ACM SIGKDD, pp.617-622, 2004.

[23] J.-G. Lee, J. Han, and K.-Y. Whang, "Trajectory Clustering: A Partition-and-Group Framework," Proc. ACM SIGMOD, pp. 593- 604, 2007.

[24]Y. Wang, E.-P. Lim, and S.-Y. Hwang, "Efficient Mining of Group Patterns from User Movement Data," Data Knowledge Eng., vol. 57, no. 3, pp. 240-282, 2006.

[25] M.Nanni and D.Pedreschi, "Time-Focused Clustering Trajectories of Moving Objects,"J. Intelli gent Information Systems, vol. 27, no. 3, pp. 267-289, 2006.

[26] L.Chen, M.Tamer O ¨ zsu, and V. Oria, "Robust and Fast Similarity Search for Moving Object Trajectories," Proc. ACM SIGMOD, pp. 491-502, 2005.

[27] V.Guralnik and G.Karypis, "A Scalable Algorithm for Clustering Sequential Data," Proc.1st IEEE Int'l Conf. Data Mining, pp. 179-186, 2001.

[28] J. Yang and W. Wang, "CLUSEQ: Efficient and Effective Sequence Clustering," Proc.19th Int'l Conf. Data Eng., pp. 101-112, Mar. 2003.

[29] A. Strehl and J. Ghosh, "Cluster Ensembles—A Knowledge Reuse Framework for combining Partitionings," Proc. Conf. Artificial Intelligence, pp. 93-98, July 2002.

[30] H. Ayad, O.A. Basir, and M. Kamel, "A Probabilistic Model Using Info Theoretic Measures for Cluster Ensembles," Proc. Fifth Int'l Workshop Multiple Classifier Systems, pp. 144-153, June 2004.

[31] A.L.N. Fred and A.K. Jain, "Combining Multiple Clusterings Using Evidence Accumulation," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 27, no. 6, pp.835-850, June 2005.

[32] X.Z. Fern and C.E. Brodley, "Random Projection for High Dimensional Data Clustering: A Cluster Ensemble Approach," Proc. 20th Int'l Conf. Machine Learning, pp. 1186-1193, June 2003.

[33] H. Kargupta, W. Huang, K. Sivakumar, and E.L. Johnson, "Distributed Clustering Using Collective Principal Component Analysis," Knowledge and Information System, vol. 3, pp. 422-448, 2001.

[34] H.T. Kung and D. Vlah, "Efficient Location Tracking Using Sensor Networks," Proc. Conf. IEEE Wireless Comm. and Networking, vol. 3, pp. 1954-1961, Mar. 2003.

[35] J. Tang, B. Hao, and A. Sen, "Relay Node Placement in Large Scale Wireless Sensor Networks," J. Computer Comm., special issue on sensor networks, vol. 29, no. 4, pp. 490-501, 2006.
[36] M. Younis and K. Akkaya, "Strategies and Techniques for Node Placement in Wireless Sensor Networks: A Survey," Ad Hoc Networks, vol. 6, no. 4, pp. 621-655, 2008.
[37] S. Pandey, S. Dong, P. Agrawal, and K. Sivalingam, "A Hybrid Approach to Optimize Node Placements in Hierarchical Heterogeneous Networks," Proc. IEEE Conf. Wireless Comm. and Networking Conf., pp. 3918-3923, Mar. 2007.
[38] Stargate: A Platform X Project, http://platformx.sourceforge.net, 2010.
[39] Mica2 Sensor Board, http://www.xbow.com, 2010.
[40] J.N. Al Karaki and A.E. Kamal, "Routing Techniques in Wireless Sensor Networks: A Survey," IEEE Wireless Comm., vol. 11, no. 6, pp. 6-28, Dec. 2004.
[41] J.Hightower and G. Borriello,"Location Systems for Ubiquitous Computing," Computer, vol. 34, no. 8, pp. 57-66, Aug. 2001.
[42] D. Li, K.D. Wong, Y.H. Hu, and A.M. Sayeed, "Detection, Classification, and Tracking of targets," IEEE Signal Processing Magazine, vol. 19, no. 2, pp. 17-30, Mar. 2002.
[43] Hsiao-Ping Tsai , De-Nian Yang, Ming-Syan Chen, "Exploring Application-Level Semantics for Data Compression", IEEE Knowledge and data engineering, vol.23, no.1, Jan. 2011.