Volume 6, Issue 9, September 2016 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Improve Dynamic Filtering for Transaction Web Dataset Using Multi Cluster Prediction Strategy Model

[1]K. M. Padmapriya, [2]G. Menaka
[1] M.Sc., M.Phil., Asst. Professor, PG & Research Department of Computer Science, SSM College of Arts and Science, Komarapalayam, Tamilnadu, India
[2] M.Phil Research Scholar, SSM College of Arts and Science, Komarapalayam, Tamilnadu, India

Abstract— *In this paper LBS systems employ a pull model or user-initiated model, where a user issues a query to a server which responds with location aware answers. To provide users with instant replies, a push model or server-initiated model is becoming an inevitable computing model in next-generation location-based services. In addition, thesis on Location-Based Service (LBS) have been emerging in recent years due to a wide range of potential applications. One of the active topics is the mining and prediction of mobile movements and associated transactions. Most of existing studies focus on discovering mobile patterns from the whole logs. However, this kind of patterns may not be precise enough for predictions since the differentiated mobile behaviors among users and temporal periods are not considered. The paper proposes a novel algorithm, namely, Cluster-based Temporal Mode Sequential Pattern Mine (CTMSP-Mine), to discover the Cluster-based Temporal Mode Sequential Patterns (CTMSPs). Moreover, a prediction strategy is proposed to predict the subsequent mobile behaviors. In ECTMSP-Mine, user clusters are constructed by a novel algorithm named Enhanced Cluster Affinity Search Technique (ECAST) and similarities between users are evaluated by the proposed measure, Alternative Location-Based Service Alignment (ALBS-Alignment). Mean while, a time segmentation approach is presented to find segmenting time intervals where similar mobile characteristics exist. To the best knowledge, this is the first work on mining and prediction of mobile behaviors with considerations of user relations and temporal property simultaneously. Through experimental evaluation under various simulated conditions, the proposed methods are shown to deliver excellent performance.*

Keywords—*Web Dataset, Filtering, Prediction Strategy, R- Tree, ALBS,*

I. INTRODUCTION

Data mining is the process of extracting patterns from data. Data mining is seen as an increasingly important tool by modern business to transform data into an informational advantage. It is currently used in a wide range of profiling practices, such as marketing, surveillance, fraud detection, and scientific discovery.

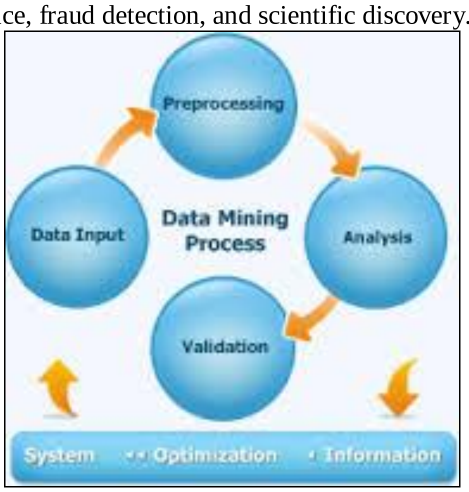

Fig 1.1 Data Mining

Data mining commonly involves four classes of tasks:

- Clustering - is the task of discovering groups and structures in the data that are in some way or another "similar", without using known structures in the data.
- Classification - is the task of generalizing known structure to apply to new data. For example, an email program might attempt to classify an email as legitimate or spam. Common algorithms include decision tree learning, nearest neighbor, naive Bayesian classification, neural networks and support vector machines.
- Regression - Attempts to find a function which models the data with the least error.

Association rule learning - Searches for relationships between variables. For example a supermarket might gather data on customer purchasing habits. Using association rule learning, the supermarket can determine which products are frequently bought together and use this information for marketing purposes. This is sometimes referred to as market basket analysis

An essential task in our proposed framework is to determine the frequent item set with the problem may be solved by using store and item category ontology. However, the store or item ontology may not match with the mobile transaction dataset. The finding problem is to automatically compute the store and item similarities from the mobile transaction dataset, which captures mobile users' moving and transactional behaviors (in terms of movement among stores and purchased items). From the database are following information available: 1) for a given store, know which items are available for sale; 2) for a given item, know which stores sell this item.

The information can help us to infer which stores or items are similar. As observe that people usually purchase similar items in certain stores, these stores may be considered as similar. The propose system is a parameter-less data mining model, named Similarity Inference Model, to tackle this task of computing store and item similarities.

In proposed system plan to explore more efficient mobile commerce pattern mining algorithm, design more efficient similarity inference models, and develop profound prediction strategies to further enhance the MCE framework. In addition, we plan to apply the MCE framework to other applications, such as object tracking sensor networks and location-based services, aiming to achieve high precision in predicting object behaviors. Therefore, proposed system develops a Enhanced Cluster-based Temporal Mobile Sequential Pattern Mine (ECTMSP-Mine) algorithm.

II. LITERATURE SURVEY

Xin Caoy et al [1] describe the location-aware keyword query returns ranked objects that are near a query location and that have textual descriptions that match query keywords. This query occurs inherently in many types of mobile and traditional web services and applications, e.g., Yellow Pages and Maps services. Previous work considers the potential results of such a query as being independent when ranking them. However, a relevant result object with nearby objects that are also relevant to the query is likely to be preferable over a relevant object without relevant nearby objects. The paper proposes the concept of prestige-based relevance to capture both the textual relevance of an object to a query and the effects of nearby objects. Based on this, a new type of query, the Location-aware top-k Prestige-based Text retrieval (LkPT) query, is proposed that retrieves the top-k spatial web objects ranked according to both prestige-based relevance and location proximity.

Xin Cao et al [2] describe a the proliferation of geo-positioning and geo-tagging, spatial web objects that possess both a geographical location and a textual description are gaining in prevalence, and spatial keyword queries that exploit both location and textual description are gaining in prominence. However, the queries studied so far generally focus on finding individual objects that each satisfy a query rather than finding groups of objects where the objects in a group collectively satisfy a query. Define the problem of retrieving a group of spatial web objects such that the group's keywords cover the query's keywords and such that objects are nearest to the query location and have the lowest inter-object distances. Specifically, we study two variants of this problem, both of which are NP-complete. We devise exact solutions as well as approximate solutions with provable approximation bounds to the problems. We present empirical studies that offer insight into the efficiency and accuracy of the solutions.

Ju Fan et al [3] descrbe a location-based services (LBS) have become more and more ubiquitous recently. Existing methods focus on finding relevant points-of-interest (POIs) based on users' locations and query keywords. Nowadays, modern LBS applications generate a new kind of spatio-textual data, regions-of-interest (ROIs), containing region-based spatial information and textual description, e.g., mobile user profiles with active regions and interest tags. To satisfy search requirements on ROIs, we study a new research problem, called spatio-textual similarity search: Given a set of ROIs and a query ROI, we find the similar ROIs by considering spatial overlap and textual similarity. Spatio-textual similarity search has many important applications, e.g., social marketing in location-aware social networks. It calls for an efficient search method to support large scales of spatio-textual data in LBS systems.

Jiaheng Lu et al [4]descrbe a geographic objects associated with descriptive texts are becoming prevalent. This gives prominence to spatial keyword queries that take into account both the locations and textual descriptions of content. Specifically, the relevance of an object to a query is measured by spatial-textual similarity that is based on both spatial proximity and textual similarity. In this paper, we define Reverse Spatial Textual k Nearest Neighbor (RSTkNN) query, i.e., finding objects that take the query object as one of their k most spatial-textual similar objects. Existing works on reverse kNN queries focus solely on spatial locations but ignore text relevance. To answer RSTkNN queries efficiently, we propose a hybrid index tree called IUR-tree (Intersection-Union R-Tree) that effectively combines location proximity with textual similarity. Based on the IUR-tree, we design a branch-and-bound search algorithm. To further accelerate the query processing, we propose an enhanced variant of the IUR-tree called clustered IUR-tree and two corresponding optimization algorithms. Empirical studies show that the proposed algorithms offer scalability and are capable of excellent performance.

Dingming Wu et al [5] describe a moving top-k spatial keyword (MkSK) query, which takes into account a continuously moving query location, enables a mobile client to be continuously aware of the top-k spatial web objects that best match a query with respect to location and text relevance. The increasing mobile use of the web and the proliferation of geo-positioning render it of interest to consider a scenario where spatial keyword search is outsourced to a separate service provider capable at handling the voluminous spatial web objects available from various sources. A key challenge is that the service provider may return inaccurate or incorrect query results (intentionally or not), e.g., due to cost considerations or invasion of hackers. Therefore, it is attractive to be able to authenticate the query results at the client side.

III. R-TREE METHODOLOGY

R-trees are tree data structures used for spatial access methods, i.e., for indexing multi-dimensional information such as geographical coordinates, rectangles or polygons. The R-tree was proposed has found significant use in both theoretical and applied contexts. A common real-world usage for an R-tree might be to store spatial objects such as restaurant locations or the polygons that typical maps are made of: streets, buildings, outlines of lakes, coastlines, etc. and then find answers quickly to queries such as "Find all museums within 2 km of my current location", "retrieve all road segments within 2 km of my location" (to display them in a navigation system) or "find the nearest gas station" (although not taking roads into account). The R-tree can also accelerate nearest neighbor search for various distance metrics, including great-circle distance.

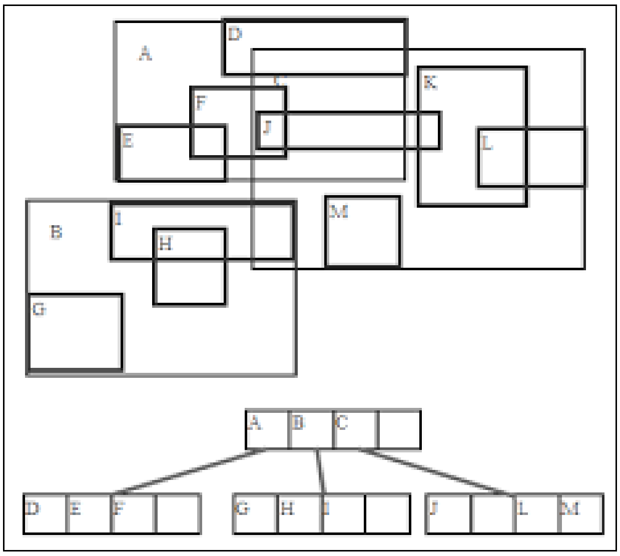

Fig 3.1 Mobile Mesh Node: A, B, C

A. R-Tree Structure

R-trees are hierarchical data structures based on B+ trees. They are used for the dynamic organization of a set of d-dimensional geometric objects representing them by the minimum bounding d-dimensional rectangles (MBR). Each node of the R-tree corresponds to the MBR that bounds its children. The leaves of the tree contain pointers to the database objects instead of pointers to children nodes. The nodes are implemented as disk pages.

An MBR can be included (in the geometrical sense) in many nodes, but it can be associated to only one of them. This means that a spatial search may visit many nodes before confirming the existence of a given MBR. Also, it is easy to see that the representation of geometric objects through their MBRs may result in false alarms. To resolve false alarms, the candidate objects must be examined. For instance, Figure 4.1 and 4.2 illustrates the case where two polygons do not intersect each other, but their MBRs do. Therefore, the R-tree plays the role of a filtering mechanism to reduce the costly direct examination of geometric objects

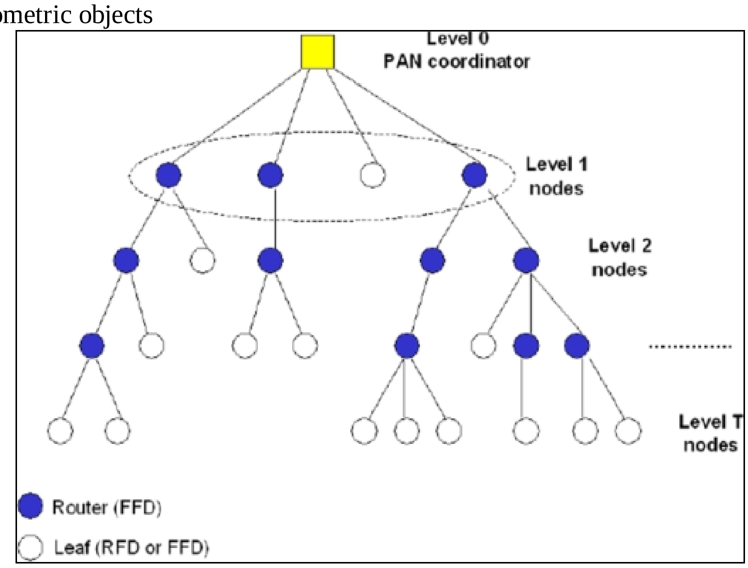

Fig 3.2 R-Trees Topology

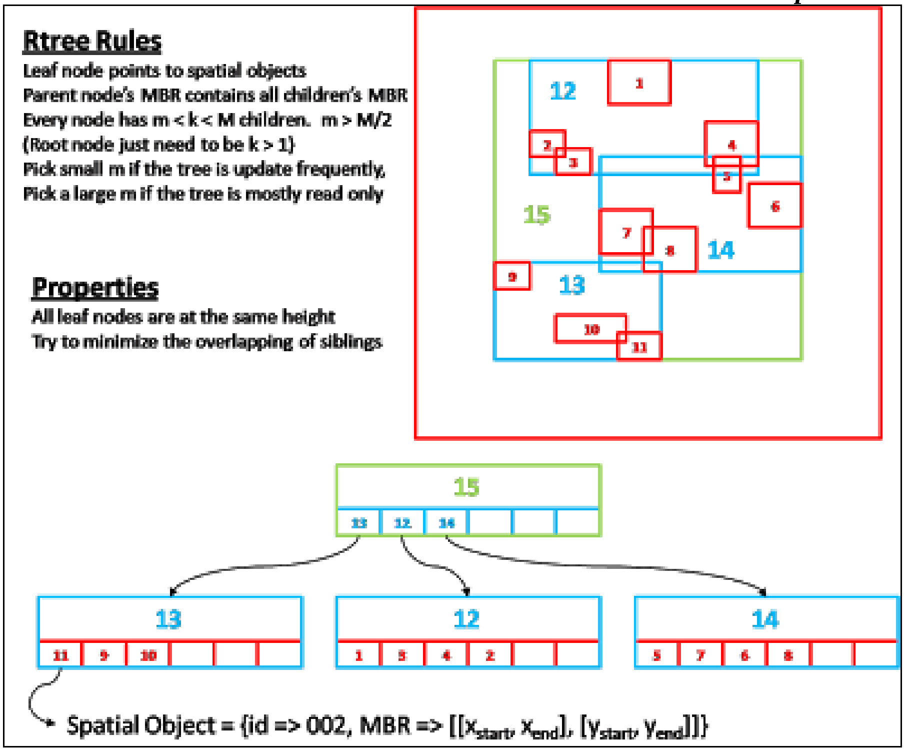

Fig 3.3 MBR TREE

An R-tree of order (m,M) has the following characteristics:

- Each leaf node (unless it is the root) can host up to M entries, whereas the minimum allowed number of entries is m ! M/2. Each entry is of the form (mbr, oid), such that mbr is the MBR that spatially contains the object and oid is the object's identifier.
- The number of entries that each internal node can store is again between m ! M/2 and M. Each entry is of the form (mbr, p), where p is a pointer to a child of the node and mbr is the MBR that spatially contains the MBRs contained in this child.
- The minimum allowed number of entries in the root node is 2, unless it is a leaf (in this case, it may contain zero or a single entry).
- All leaves of the R-tree are at the same level.

From the definition of the R-tree, it follows that it is a height-balanced tree. As mentioned, it comprises a generalization of the B+-tree structure for many dimensions. R-trees are dynamic data structures, i.e., global reorganization is not required to handle insertions or deletions. Figure 4.3 shows a set of the MBRs of some data geometric objects (not shown). These MBRs are D,E, F,G,H, I, J,K,L,M, and N, which will be stored at the leaf level of the R-tree. The same figure demonstrates the three MBRs (A,B, and C) that organize the aforementioned rectangles into an internal node of the R-tree. Assuming that M = 4 and m = 2, Figure 4.4 depicts the corresponding MBR. It is evident that several R-trees can represent the same set of data rectangles. Each time, the resulting R-tree is determined by the insertion (and/or deletion) order of its entries Fig 4.5.

```
Algorithm RangeSearch (TypeNode RN, TypeRegion Q)
/* Finds all rectangles that are stored in an R-tree with root node RN, which are
intersected by a query rectangle Q. Answers are stored in the set A */
if RN is not a leaf node
        examine each entry e of RN to find those e.mbr that intersect Q
foreach such entry e call RangeSearch(e.ptr,Q)
else // RN is a leaf node
        examine all entries e and find those for which e.mbr intersects Q
        add these entries to the answer set A
endif
Insert (TypeEntry E, TypeNode RN)
/* Inserts a new entry E in an R-tree with root node RN */
```

Step 1:

Traverse the tree from root RN to the appropriate leaf. At each level, select the node, L, whose MBR will require the minimum area enlargement to cover E.mbr

Step 2:

In case of ties, select the node whose MBR has the minimum area
if the selected leaf L can accommodate E

Insert E into L
Update all MBRs in the path from the root to L,so that all of them cover E.mbr
Else // L is already full

Step 3:

Let E be the set consisting of all L's entries and the new entry E Select as seeds two entries e1, e2 ! E, where the distance between e1 and e2 is the maximum among all other pairs of entries from E Form two nodes, L1 and L2, where the first contains e1 and the second e2

Step 4:

Examine the remaining members of E one by one and assign them to L1 or L2, depending on which of the MBRs of these nodes will require the minimum area enlargement so as to cover this entry
if a tie occurs
Assign the entry to the node whose MBR has the smaller area
endif
if a tie occurs again
Assign the entry to the node that contains the smaller number of entries
endif

Step 5:

if during the assignment of entries, there remain " entries to be assigned and the one node contains m " " entries

Step 6:

Assign all the remaining entries to this node without considering the aforementioned criteria /* so that the node will contain at least m entries */ endif

Step 7:

Update the MBRs of nodes that are in the path from root to L, so as to cover L1 and accommodate L2

Step 8:

Perform splits at the upper levels if necessary
In case the root has to be split, create a new root
Increase the height of the tree by one
endif

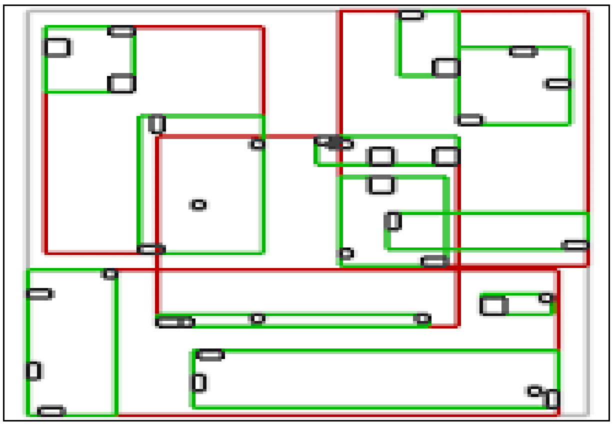

Fig. 3.4 Data MBRs and their MBRs

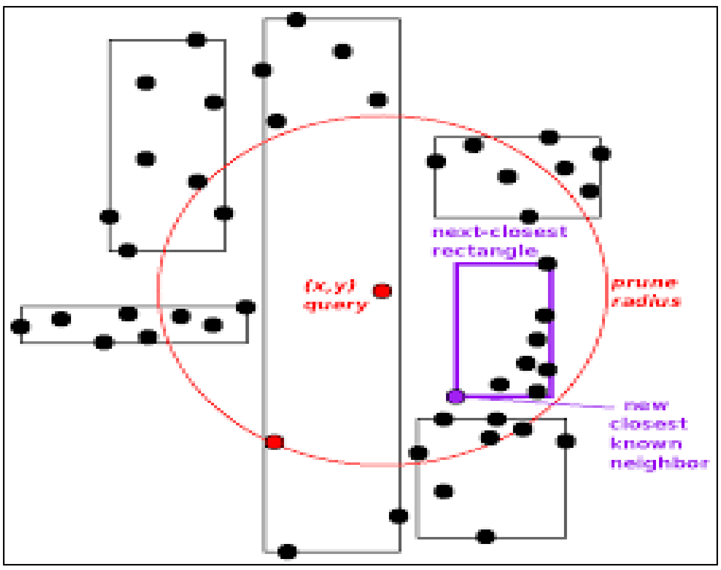

Fig. 3.5 Data with Corresponding R-tree

IV. ENHANCED CLUSTERING OF MOBILE TRANSACTION SEQUENCES

The PMCP-Mine algorithm is performed in a bottom-up manner. We first discover frequent transaction behaviors in a single store, eventually, the complete mobile commerce patterns can be obtained by the PMCP-Mine algorithm. The PMCP-Mine algorithm is divided into three main phases:

In order to mine the cluster-based temporal mobile sequential patterns efficiently, we proposed a novel method named CTMSP-Mine to achieve this mining procedure. In ECTMSP-Mine, both factors of user cluster and time interval are taken into account such that the complete mobile sequential patterns can be discovered. The entire procedures of ECTMSP-Mine algorithm can be divided into three main steps:

1) Frequent-Transaction Mining,
2) Mobile Transaction Database Transformation, and
3) ECTMSP Mining.

- Frequent-Transaction Mining
  - In this phase, the frequent transactions (F-Transactions) are mined in each user cluster and time interval by applying a modified Apriori algorithm.
- Mobile Transaction Database Transformation
  - In this phase, F-Transactions are used to transform each mobile transaction sequence S into a frequent mobile transaction sequence S'. According to Table 3, if a transaction T in S is frequent, T would be transformed into the corresponding F-Transaction. Otherwise, the cell of T would be transformed into a part of path.

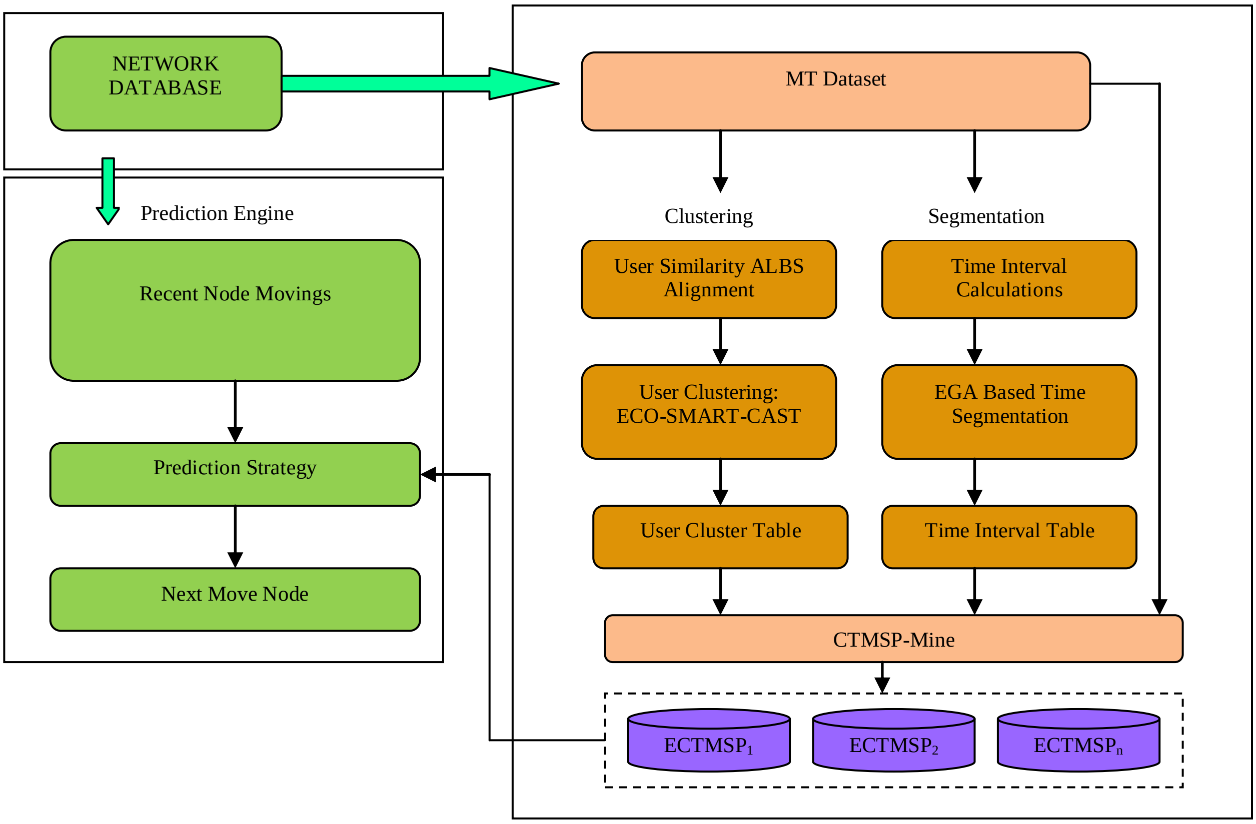

Fig 4.1 Enhanced Clustering Of Mobile Transaction Sequences Dataset

A mobile transaction sequence can be viewed as a sequence string, where each element in the string indicates a mobile transaction. The major challenge to tackle is to measure the content similarity between mobile transactions. The ALBS-Alignment algorithm is proposed, which can obtain the similarity. ALBS-Alignment is based on the consideration that two mobile transaction sequences are more similar, when the orders and timestamps of their mobile transactions are more similar. ECAST algorithm is used to cluster the users. In a mobile transaction database, users in the different user groups may have different mobile transaction behaviors. The first task to tackle is to cluster mobile transaction sequences. In this module, a parameter-less clustering algorithm called ECAST is proposed.

Before performing the ECAST, a similarity matrix S is to be generated, based on the mobile transaction database. The entry Sij in matrix S represents the similarity of the mobile transaction sequences i and j in the database, with the degrees in the range of [0, 1].

A mobile transaction sequence can be viewed as a sequence string, where each element in the string indicates a mobile transaction. The major challenge to tackle is to measure the content similarity between mobile transactions. The proposed system has following advantages,

- Predicts the subsequent user mobile behaviors effectively.
- Generate the most suitable time intervals for time segmentation.

- Mines and predicts the mobile behaviors with considerations of user relations and temporal property simultaneously.
- Suitable for Location-Based Service Environments.

A. LBS-Alignment Algorithm

The LBS-Alignment algorithm is proposed, which can obtain the similarity. LBS-Alignment is based on the consideration that two mobile transaction sequences are more similar, when the orders and timestamps of their mobile transactions are more similar.

Based on this concept, it is designed specifically the time penalty (TP) and the service reward (SR) in the ALBS-Alignment. The base similarity score is set as 0.5. Two mobile transactions can be aligned if their locations are the same. Otherwise, a location penalty is generated to decrease their similarity score. The location penalty is defined as 0.5/ ($|s_1|$ + $|s_2|$) where $|s_1|$ and $|s_2|$ are the lengths of sequences s_1 and s_2 , respectively. Notice that the maximal number of location penalties is $|s_1| + |s_2|$. When two sequences are totally different, their similarity score is 0.

When two mobile transactions are aligned, the time penalty and service reward are measured. TP focuses on their time distance. The farther the time distances between them, the larger their time penalty. TP that is generated to decrease their similarity score is defined as (($|s_1$ time – s_2 time $|$)/ len, where len indicates the time length. SR focuses on the similarity of the service requests. The more similar their service requests, the larger their service reward.

SR that is generated to increase their similarity score is defined as |s1.services Q□s_2.services|/|s_1.services Q□s_2.services|. The following algorithm shows the procedures of an LBS-Alignment measure. Input data include two mobile transaction sequences. Output data are the similarity between two mobile transaction sequences, with the degrees in the range from 0 to 1. Some parameters are initialized. The base similarity score is set as 0.5.

Dynamic programming is used to calculate Mi;j (line 6 to line 16).M*i,j* indicates the value of matrix M in column i and row j, where M is the score matrix of ALBS-Alignment. In this procedure, if the locations of two transactions are the same (line 8), both the time penalty (line 9) and the service reward (line 10) are calculated to measure the similarity score (line 11). Otherwise (line 12), the location penalty is generated to decrease the similarity score (line 13). Finally, $M_{s.length,s'.length}$ is returned as the similarity score of the two mobile transaction sequences (line 17).

/* ALSB Algorithm */

Input: Two mobile transaction sequences s and s’
Output: The similarity between s and s’
01 ALBS-Alignment (s,S’)
02 p ← 0.5 / (s.length + s’.length) /*p is the location penalty */
03 $M_{0,0}$ ← 0.5
04 $M_{i,0}$ ← $M_{i-1,0}$ - p□ □□ i = {1,2,....,s.length)
05 $M_{0,j}$ ←$M_{0,j-1}$ - p□ □□j = {1,2,....,s’.length)
06 For i ←1 to s.length
07 For j ←1 to s’.length
08 For j ←1 to s’.length
09If si.location = sj’.location
10 TP ←p * | si.time = sj’.time| / len /* time penalty */
11 SR ←p * (si.service ∩□sj.’service/ si.service U □sj.’service)
/* service reward */
12 $M_{i,j}$ ←Max($M_{i-1,j}$-1 – TP + SR, $M_{i+1,j}$ – p, $M_{i,j}$-1 – p)
14 Else
15$M_{i,j}$ ←Max($M_{i-1,j}$– p, $M_{i,j}$-1 – p)
16 End If
17 End For
18 End For
19 Return $M_{s.length,s'.length}$

B. ECAST Algorithm

After obtaining the similarity matrix, the mobile transaction sequences is clustered by the proposed E-CO-Smart-CAST. The following ECO-SMART-CAST algorithm shows the procedure of ECO-Smart-CAST. The input data are an N-by-N similarity matrix. The output data are the clustering result. ECO-Smart-CAST can automatically cluster the data according to the similarity matrix without any user-input parameter. The main ideas of ECO-Smart-CAST are as follows: First, the ECAST method that takes a parameter named affinity threshold t is used as the basic clustering method.

Second, a quality validation method is used, called Hubert’s T Statistics, to find the best clustering result. Third, a hierarchical concept is used to reduce the sparse clusters. For a clustering result, Hubert’s T Statistics is used to measure its quality by taking the similarity matrix and the clustering result as the input. In each clustering result, its Tobj and Tclu is calculated which represent the clustering qualities measured by the original object similarity matrix S and the last cluster similarity matrix S’, respectively. The initial values of S’ and S are the same since to let every object be an independent cluster (line 4).

To determine the most suitable t, the easiest way is varying t with a fixed increment and iterating the executions of ECAST to find the best clustering result. The main drawback of this way is that many iterations of computation are required. For this reason, the number of computations is reduced by eliminating unnecessary executions, and then, obtain a “near-optimal” clustering result. That is a minimal number of ECAST executions is performed.

The main idea is to narrow down the range of t effectively. A testing range R for setting t is from 0 to 1. (Line 3). By the points P0, P1, P2, P3, and P4, R is equally divided into five points, where P0 <P1 <P2 <P3 <P4. Then, the value of each Pi (line 6) is sequentially taken as the affinity threshold to perform the CAST algorithm (line 7). Finally, the clustering result with the highest quality during the tested process is returned (line 8).

/* ECAST Algorithm */
Input: An N-by-N similarity matrix S
Output: The clustering result
1 ECO-Smart-CAST (S)
2 Clustering_Result ← PI, T_{CO_Best} tends to -1,S’ □ S
3 R ← [0,1] / * R_{upper} is 0, R_{lower} is 1 */
4 Do
5 For I ←0 to 4
6 P_i, ←I * (R_{upper} – R_{lower}) / 4 + R_{lower}
7 CR_i ←CAST(Clustering_Result,S’,P_i)
8 Return CRi

V. RESULTS AND DISCUSSION

The following **Table 5.1** describes experimental result for R-Tree and CTMSP Tree error rate analysis. The table contains number of transaction datasets with count and error rate analysis for R-Tree and CTMSP Tree rate deatils are shown.

Error Rate = ([R1-R2]/R1) * 100

S.NO	NUMBER OF Transcation datasets [R1]	Error Rate Analysis			
		R-Tree Tracking Topology		CTMSP Topology	
		Tracking Node [R2]	Error Rate [%]	Tracking Node [R2]	Error Rate [%]
1	100	70	30	75	25
2	200	150	25	162	19
3	300	253	15	265	12
4	400	324	19	341	15
5	500	467	07	469	06
6	600	585	02	594	01
7	700	645	08	658	06
8	800	788	01	793	08
9	900	853	05	876	03
10	1000	943	06	953	05

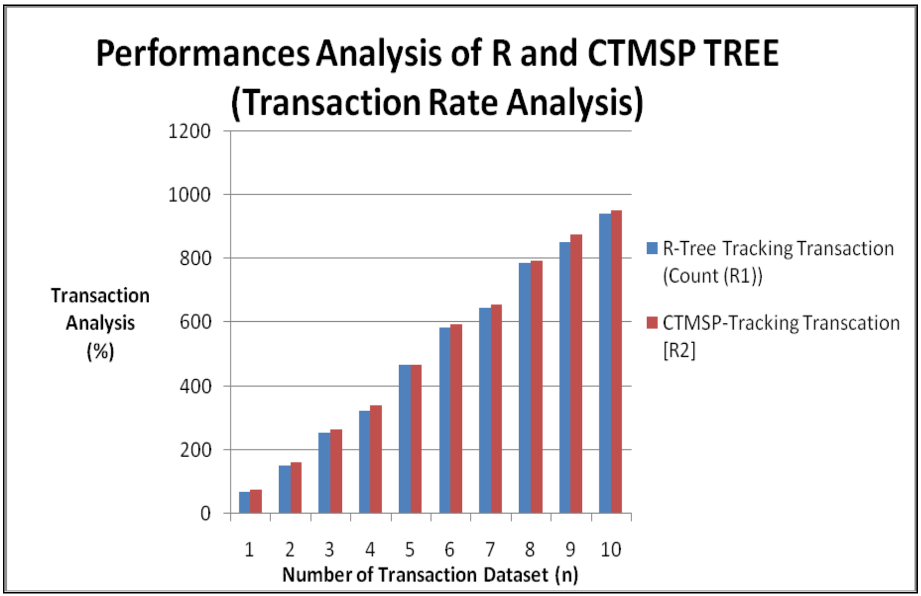

Fig 5.1 Performances Transaction Analysis of R-CTMSP Tree

The following Fig 5.1 describes experimental result for R-Tree and CTMSP Tree Transcation rate analysis. The figure contains number of transaction rate for R-Tree and CTMSP Tree details are shown. The following Fig 5.2 describes experimental result for R-Tree and CTMSP Tree error rate analysis. The figure contains AVG error rate for R-Tree and CTMSP Tree details are shown.

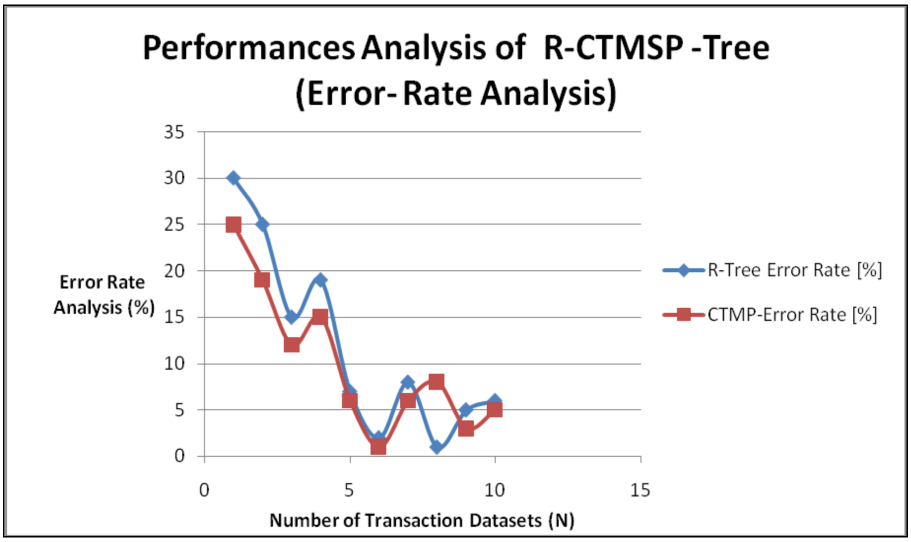

Fig 5.2 Performances Error Rate Analysis –R-Tree and CTMSP-Tree Topology

This paper is used to eliminate time complexity rate while finding high frequent item sets in a transaction database. In this proposed research, tree construction process using two strategies, namely E-MTS (Enhanced Mobile Transaction Set items) and LSB-CMSTP. It also used to reduce number scans to the database and time reduces.

VI. CONCLUSION

In this thesis, a novel method named ECTMSP-Mine is proposed, for discovering CTMSPs in LBS environments. Furthermore, novel prediction strategies are proposed to predict the subsequent user mobile behaviors using the discovered ECTMSPs. In ECTMSP-Mine, first a transaction clustering algorithm is proposed named ECO-Smart-CAST to form user clusters based on the mobile transactions using the proposed ELBS-Alignment similarity measurement. Then, the time segmentation algorithm is utilized to generate the most suitable time intervals. To our best of mobile behaviors associated with user clusters and temporal relations.

A series of experiments were conducted for evaluating the performance of the proposed methods. The experimental results show that ECO-Smart-CAST method achieves high-quality clustering results and the proposed ECBSS strategy obtains highly precise results for user classification. Meanwhile, the algorithms obtain the most proper and correct time intervals. For behavior prediction, ECTMSP is shown to outperform other prediction methods in terms of precision and F-measure. The experimental results demonstrate that the proposed methods are efficient and accurate under various conditions.

The application works well for given tasks in windows environment. Any node with .Net framework installed can execute the application and identifies the best site. The underlying mechanism can be extended to any / all kind of web servers and even in multi-platform like Linux, Solaris and more. The system is planned to extend the services can be given as input to IBM architecture also. The system eliminates the difficulties in the existing system. It is developed in a user-friendly manner. The system is very fast in applying algorithm. This software is very particular in predict the subsequent mobile behaviors.

- In future work, the method can be applied to real data sets. In addition, the CTMSP-Mine can be applied to other applications, such as GPS navigations, with the aim to enhance precision for predicting user behaviors.
- The application if developed as web site, can be used from anywhere.
- The new system is designed such that those enhancements can be integrated with current modules easily with less integration work.

REFERENCES

[1] X. Cao, G. Cong, and C. S. Jensen, "Retrieving top-k prestige-based relevant spatial web objects," Proc. VLDB Endowment, vol. 3, no. 1, pp. 373–384, 2010.

[2] X. Cao, G. Cong, C. S. Jensen, and B. C. Ooi, "Collective spatial keyword querying", in Proc. ACM SIGMOD Int. Conf. Manage.Data, 2011, pp. 373–384.

[3] J. Fan, G. Li, L. Zhou, S. Chen, and J. Hu, "Seal: Spatio-textual similarity search," Proc. VLDB Endowment, vol. 5, no. 9, pp. 824–835, 2012.

[4] J. Lu, Y. Lu, and G. Cong, "Reverse spatial and textual k nearest neighbor search," in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2011, pp. 349–360

[5] D. Wu, M. L. Yiu, C. S. Jensen, and G. Cong, “Efficient continuously moving top-k spatial keyword query processing,” in Proc. Int. Conf. Data Eng., 2011, pp. 541–552.

[6] B. Yao, F. Li, M. Hadjieleftheriou, and K. Hou, “Approximate string search in spatial databases,” in Proc. Int. Conf. Data Eng., 2010, pp. 545–556.

[7] D. Zhang, B. C. Ooi, and A. K. H. Tung, “Locating mapped resources in web 2.0,” in Proc. Int. Conf. Data Eng., 2010, pp. 521–532.

[8] R. Zhong, J. Fan, G. Li, K.-L. Tan, and L. Zhou, “Location-aware instant search,” in Proc. 21st ACM Int. Conf. Inf. knowl. Manage., 2012, pp. 385–394.

