Australian Journal of Basic and Applied Sciences, 10(2) Special 2016, Pages: 250-261

AUSTRALIAN JOURNAL OF BASIC AND APPLIED SCIENCES

ISSN:1991-8178 EISSN: 2309-8414
Journal home page: www.ajbasweb.com

Tree Based Space Partition of Trajectory Pattern Mining For Frequent Item Sets

[1]P.Geetha and [2] E. Ramaraj

[1]*Ph.D scholar, Alagappa University.*
[2]*Department of Computer Science and Engineering , Alagappa University, Tamil Nadu, India.*

Address For Correspondence:
P.Geetha, Ph.D scholar, Alagappa University.

ARTICLE INFO

Article history:
Received 04 December 2015
Accepted 22 January 2016
Available online 14 February 2016

Keywords:
Clustering, Data Reduction, Execution Time, Frequent Mining, Heap Tree, Memory Consumption, Space Partition

ABSTRACT

Pattern discovery from the large data base is an interdisciplinary field in a computer science termed data mining. The prediction of spatial trajectory locations of these patterns is an attractive research area leads to the evolution of frequent mining algorithms. The existence of the diverse databases degrades the execution time of the mining process. To overcome this issue, Tree based Space Partition of Trajectory Pattern Mining (TSPTPM) is proposed in this paper. A novel method, which adopts clustering and tree structure to extract the frequent patterns from the real time diverse datasets. Initially, the clustering process organizes the number of transactions into group and assigns the ID to each group. Then, heap tree algorithm constructs a tree from the cluster of transactions. In a tree structure, the transaction with maximum ID consider as the root node and the odd and even number of transactions consider as child nodes. Then, pre-order traversal extracts the frequent patterns in the tree structure. Finally, Vague Space Partition (VSP) algorithm applied to offer the flexible spatial partition of tree structure and convert into sequences. The proposed TSPTPM enhances the frequent pattern extraction process by reducing time consumption. The optimized patterns are obtained from the heap tree structure and the obtained results are effectively partitioned by using the novel VSP algorithm. The mining time analysis for various datasets namely, Chess, Mushroom, Connect and accident with various number of items in cluster. The comparative analysis of mining time for proposed TSPTPM algorithm and existing Apriori proves the effectiveness.

INTRODUCTION

Transaction Data base (TD) is an extension of frequent item set mining in large static of data mining field. The dynamic and continuous evolving nature of data base requires up-to-date analysis. The techniques such as hMinor algorithm, hCount and lossy count in the literature works reused the frequent items by the combinatorial explosion of patterns. Fixed window length and decay factor are required to implement the explosion model. The scanning and the support evaluation for item set are fast. Hence, the bit table and index bit table are applied to govern the scanning with fixed bit size. But, the storage of bit vectors and time consumption are more due to the large size database. The selection of length and decay factor values for every item sets are difficult and the memory and time consumption are more. To overcome these problems, max frequency measure varies the window length for each item sets. The extension of varying window based frequent mining to image classification methods, large uncertain database and the sub-graphs creation.

The evolution of Graph Based Mining (GBM) algorithms in frequent trajectory pattern analysis consume large search space. To reduce the search space, GBM utilizes the adjacency property between the extracted patterns. Mapping graph and transaction item sets in GBM provide the extension to small number of patterns with maximum candidate generation. The simultaneous inclusion of spatial and temporal attributes discovers the

To Cite This Article: P.Geetha and E. Ramaraj, Tree Based Space Partition of Trajectory Pattern Mining For Frequent Item Sets. ***Aust. J. Basic & Appl. Sci.,*** *10(2):250-261, 2016*

trajectory patterns without generating candidates. The trajectory pattern analysis extends its application to real time pedestrian in the form of interesting patterns extraction. The prediction of next movement of users is difficult in real time analysis. Novel cluster based prediction model evaluates the user's location on the basis of frequent behavior. The behavior based applications requires activity monitoring. Video monitoring equipment such as digital cameras are used for activity monitoring. Monitoring by the video equipment contains the following limitations:

- Pre-defined target trajectories
- Difficult in monitoring of regions other than the trajectory region
- Hard in automatic analysis and irregular activities
- Expensive digital cameras

Hence, series of cameras are replaced by an array of Radio Frequency (RF) tags. The capture of frequent trajectory pattern effectively predicts the anomalies. The inclusion of clustering and sequential mining creates an inter active cluster patterns. The approximate trajectory location increases the complexity of mining tasks. An alternative approach called crisp space partition reduces the complexity. But, sharp boundary problem is occurred. A Vague Space Partition (VSP) solves the sharp boundary problem effectively. The problems addressed in trajectory pattern approach are more memory and time consumption.

The creation of maximum similar patterns significantly reduces the testing time and memory requirement. The algorithms are grouped into two categories Breadth First Search (BFS) and Depth First Search (DFS). Apriori, partition, FP growth and intersection algorithms are available for effective pattern creation but, the density of created bit vectors is less which leads to poor compression and compaction. To increase the density, bit mask selection algorithm introduced in frequent mining.

The increase in pattern size will produce the following problems in GBM approaches.

- Computational bottleneck- the time taken for an algorithm to complete the mining process is more.
- Applicable patterns- Huge mining results maximizes the potential usage of patterns in real time applications.

The evolution of sequential pattern methods reduces the above problems. But, the most of identified patterns from sequential method are not informative. Hence, extraction of utility based patterns requires a special attention in the frequent mining domain. The introduction of USpan algorithm is to retrieve the utility based sequential pattern. The existence of tree based methods such as Sequential Pattern Trees (SPT), Maximum Utility Growth tree (MU-Growth) and Linear Prefix (LP-tree) for frequent mining perform restructuring by single pass. The problems addressed in the traditional frequent mining works are the prediction failure of meaningful patterns due to the falling of close trajectory locations into partitioned regions called sharp boundary problem. The Space Partition (SP) approach effectively solve the sharp boundary problem. The existing frequent mining works affected by more run time, memory consumption and search space. To overcome these issues, the Tree based Space Partition of Trajectory Pattern Mining (TSPTPM) algorithm proposed in this paper. The novelty of TSPTPM lies in two ways as follows:

- During tree formation, the transaction with maximum ID considered as root node. The construction of tree is based on the priority principle such that already extracted frequent pattern is not considered as root or child node rather than the node having next highest priority next to already exist pattern acts as a root node in the successive process.
- The position estimation of frequent patterns in space partition algorithm modified in order to reduce the exponential records which increases the density of sequences and assures the sufficient data compression.

The contributions of proposed TSPTPM algorithm are as follows:

- The construction of heap tree on the basis of constant time compared to other tree structures which are in sub linear time basis.
- Polynomial run time reduction by a heap structure improved the efficiency of the system.
- The application of Vague Space Partition (SP) algorithm raised up the density of sequences which reduces the scanning time considerably.

Related work:

Mining of frequent item sets considered as a problem due to the rapid arrival of large diverse databases and raised the complex storage. Stream mining algorithms were involved on the basis of approximations. The maximum frequency measure based mining considered as an effective items mining. Calders *et al.* (2014) extended mining of items into mining of item sets by using an optimized incremental algorithm. The maintenance of compact memory by the connection establishment between results from number theory and the size. The evolution of bit table and index bit table mechanisms in frequent mining of item sets utilized the fixed size bit vector for each item in a sets which increased the memory consumption and computation time. Vo *et al.* (2011) presented the Dynamic Bit Vector (DBV) approach based tree structure for frequent mining. Vo *et al.* (2012) extended the DBV approach with the lookup table approach and the interactions between the two DBVs are estimated. The assumption for memory and time preserving was discussed. The sequential patterns inside a transaction and inter-transaction pattern was mined from several datasets. Vo *et al.* (2012) utilized DBV approach for Inter Sequence Pattern (ISP) mining. Vo *et al.* (2012) proposed the concept of Most Generalization Association Rules (MGAR) on the basis of some theorems. The redundancy of the rules were checked by the hash tables. The increase in number of Frequent Closed Item sets (FCI) increased the time for generating MGAR. Vo *et al.*(2013) proposed new algorithm based on lattice structure for reduction of MGAR. The maintenance of frequent item set lattice structure is difficult. Vo *et al.* (2014) used pre-large concept in order to reduce the difficulty in maintenance. The application of frequent mining of item sets extended to image classification tasks. Fernando *et al.* (2012) referred the transaction patterns as a Frequent Local Histograms (FLH) by considering the all histogram information in mining process. The selection of required FLH patterns was performed. The extraction of frequent item sets from the uncertain database was difficult since, they contains the exponential number of words. Wang *et al.* (2012) proposed an incremental mining algorithms for the reduction of re-execution of mining algorithm for new data base. The discovery of frequent item sets from uncertain data base also performed. The item sets categorized into two namely, structured and semi-structured. The evolution of graph based representations suitable for these models. Jiang *et al.* (2013) discussed the survey of mechanisms for the generation of candidate sub-graphs and identification of desired frequent sub-graphs. The search space for construction of graph models was more.

Lee *et al.* (2009) proposed an efficient Graph Based Mining (GBM) algorithm for spatial-temporal database. The sequential processes such as generation of mapping graph and Trajectory Information (TI) lists. The mining of frequent patterns from the database performed by using Depth First Search (DFS) traversal. Research works paid an immediate attention to mobile user movement prediction. On the basis of semantic and geographic features, Ying *et al.* (2011) presented the novel cluster based prediction model for prediction of next location of mobile user's. The user behavioral models evolved in the real time process required the updating in frequent mining of item sets. Cheng *et al.* (2014) identified the computational burden of graph mining methods. Thereby, the exponential growth of result was avoided and the applicability of graph patterns were improved. Ge *et al.* (2012) presented the real time vision based analysis in pedestrian model as an example for human collecting behavior. The utilization of multiple cameras on the large field provided the various image patterns. Hence, frequent mining of image patterns played a significant role in activity monitoring process leads to critical data mining. Liu *et al.* (2012) used the Radio Frequency (RF) tag arrays for the development of practical fault tolerant method. They counteract the RF tag noise and created the models for regular activities. The large size of applications introduced the spatial-temporal mining problem, since they have approximate spatial trajectory locations. The space partition approach effectively handled the approximate nature. But, the sharp boundary problem occurred during the implementation of space partition approach. Wang *et al.* (2013) used Vague Space Partition (VSP) approach to reduce the sharp boundary problem. They divided the spatial plane into set of vague space grid cells and transformed the trajectory locations into cells by using distance based membership function. The inclusion of clustering and sequential mining improved the split the trajectory pattern area. Shaw and Gopalan (2014) applied the threshold for the extraction of active clusters and they were arranged in descending order on the basis of number of passed trajectories. The increase in volume of data affected the memory consumption and testing time adversely.

Basu and Mishra (2010) developed an efficient bit masking search algorithm for the test data in order to create maximum similar patterns. The bit mask based compression on the basis of dictionary selection method substantially reduced the memory and time consumption. The bit mask algorithms were based on the concept of dictionaries. The run time analysis of association rule mining algorithms such as trees, hashing and depth first search and bit mask increased. Venkatesan and Ramaraj (2011) used bit search technique effectively reduced the search and memory space there by efficiency was increased. An Apriori algorithm in rule mining algorithm was popularly used. But, some limitations such as several iterations for data mining, irrelevant items generation and prediction difficulties of unusual events were addressed in the implementation of Apriori algorithm. Abaya (2012) provided the improvement in Apriori algorithm in such a way that the elimination of non-significant candidate keys by the introduction of factors of set size and frequency. The traditional bit search technique increased the projection length and not compatible with CPU 32 bit performance.

Boaddh *et al.* (2012) increased the density of bit vectors in order to satisfy the CPU 32 – bit performance by bit mask search algorithm. The bit mask search representation improved the compression and compaction there by efficiency of frequent mining improved. The making of strong association rules for frequent mining was difficult and the scalability with the document size also required. Boaddh *et al.* (2012) proposed an alternative approach refers the bit search technique to reduce the scanning time. They utilized trees, arrays, hashing, tree based approaches to validate the run time analysis. Cameron *et al.* (2014) introduced the bit wise parallel algorithms for the extraction of matched contents. The usage of large number of entities degraded the mining parameters such as search space and execution time. Geetha and Raj (2014) considered UP growth algorithm for mining of huge entities. Also, the discussion about the application of Binary Mask search algorithm for handling the density of bit vectors. The evolution of modern processors by the multiple cores provided the maximum power output. Zhang *et al.* (2011) reviewed the exploitation of Single Instruction Multiple Data (SIMD) process for the improvement of tree search performance. The overhead in memory access was reduced by dynamic blocking pre-fetch on the basis of blocked tree architecture. Sequential pattern mining involved in consumer behavior analysis of business concern. The enhancement in pattern selection framework provided by mining of high utility patterns (frequent or infrequent).

Yin *et al.* (2012) presented an USpan algorithm to extract the high utility patterns. They introduced the lexicographic quantitative sequence tree for the calculation of node utility with pruning strategies. Sequential pattern methods were not suitable for large classifier banks. Hence, Ong *et al.* (2012) used the sequential pattern trees for multi-class classifiers in a sign language recognition. On the basis of hand trajectories, they built sign level classifiers from sub units. Each item in the item sets have its own significance. Vo *et al.* (2013) addressed the significance by proposal of weighed items transaction databases. Hence, mining of Frequent Weighed Item sets (FWI) presented by an algorithm of Weighed Item sets Tree (WIT). Traditional tree based algorithms created the nodes in an independent manner. The construction of tree based on the connection establishment by the pointers. The involvement of number of pointers leads to inefficient frequent mining. Pyun *et al.* (2014) presented the new form of tree referred Linear Prefix Tree (LPT) in contains array forms thereby pointers between the nodes were minimized. Huge amount of real time applications evolved high utility item sets. Hence, mining of those sets were considered as an important issue in the research areas. Yun *et al.* (2014) presented an algorithm termed as Maximum Utility Growth (MU-Growth), which used the Maximum Item Quantity Tree (MIQ). MIQ captured the information about the database with single pass and the restructuring of MIQ effectively reduced the overestimated utilities. The MIQ based methods outperformed with the traditional overestimated methods for the parameters of runtime constraints. But, the execution time was more in the traditional approaches. Hence, frequent mining time reduced by using the proposed method of Tree based Space Partition of Trajectory Pattern Mining (TSPTPM).

Tree Based Space Partition of Trajectory Pattern Mining:

This section illustrates the implementation of proposed Tree based Space Partition of Trajectory Pattern Mining (TSPTPM) search algorithm in sequential processes as follows:

1. Clustering
2. Heap Tree formation
3. Tree based search
4. Vague Space Partition

The flow diagram of proposed method is shown in Fig. 1. Initially, the dataset is passed through the clustering process. The formation of clusters based on the user choice. For example, the user choice is 1000, then the transactions in accident dataset group the 1000 transactions in a single cluster. Similarly 30 clusters of each 1000 entries are formed. The remaining entries are included in another cluster. These clusters proceeded in forthcoming stages. Each cluster is applied to heap tree formation. In that, transaction with maximum ID considered as root node, then, the nodes represents even number of transactions regarded as left nodes and the nodes with odd number of transactions as right nodes to the root node. The tree based transaction data base is applied to the Vague Space Partition (VSP) algorithm which performs scanning process. The time interval between start, end and support are calculated. The vague space partition based on prefix span is used in this paper to extract the frequent mining of given dataset reduces the computational time effectively.

3.1Dataset:

An open source data mining library termed as SPMF offers various item sets, utility item sets for frequent mining. There are six datasets are collected from the http://fimi.cs.helsinki.fi/data/ such as chess, mushroom, connect and accident. The transactions and the number of distinct items for various datasets listed in Table 1.

Table 1: Dataset description

Dataset Name	Number of Transactions	Number of distinct items
Chess	3196	76
Mushroom	8124	120
Connect	67,557	130
Accidents	340184	468

The successive processes are performed in proposed TSPTPM algorithm as follows:

Step 1: Initialize the transactions from the dataset(Tr)

Step 2: Formation of clusters $[C_n] = clustering(Tr_n)$

Step 3: Construction of tree $(T_H) = Heap\ tree(C_i)$

Step 4: Application of space partition approach to the formed tree $patt_F = space\ partition(T_H)$ and produce output

3.2. Clustering:

The first and foremost process in proposed TSPTPM algorithm is clustering. In this process, the transactions from the data sets are extracted and grouped together as a cluster of each transaction with the ID. The incremental clustering algorithm to form the cluster is as follows: Initially, the transactions from the datasets are extracted and stored in Tr_n . The initialization phase includes the cluster centroid, counter initialization. For each value of transactions, the new value of cluster is computed from the comparison between the centroid and the threshold value. The ID is assigned for newly added transactions to the clusters.

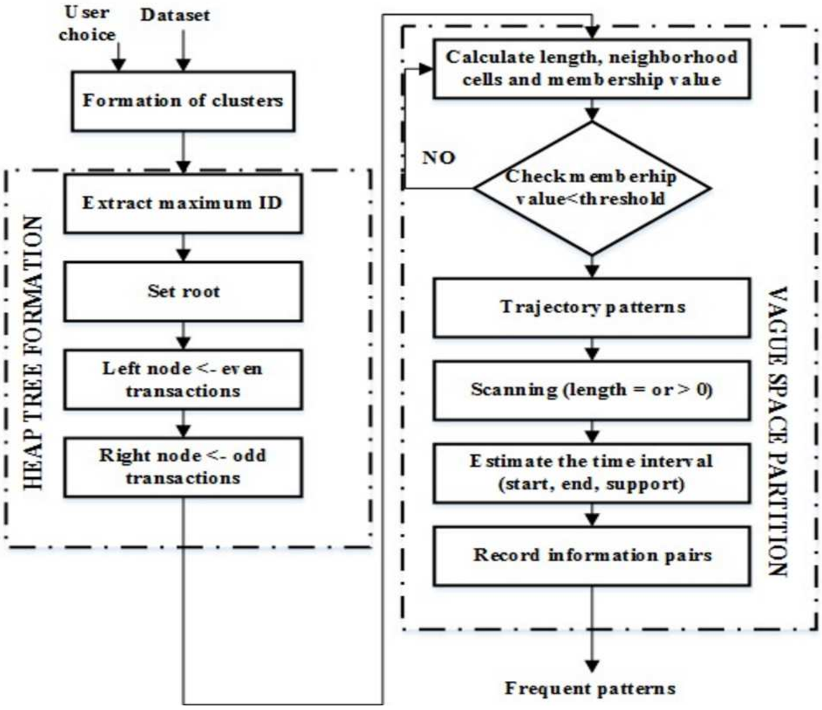

Fig. 1: Flow diagram of proposed TSPTPM algorithm.

Clustering

Input: *Dataset (D), number of clusters(K), threshold*

Output: *Clusters(C_n)*

1. *Initialize the clusters as null*
2. *Initialize the transactions as (Tr_n)=transactions of (D)*
3. *Set centroid of cluster as x_i*
4. *Initialize the counter as count=1.*
5. *For all $x_i \in Tr_n$*
6. *For all Cluster$\in$ $Clusters$*
7. *If $\|x_i - centroid(Cluster)\| < threshold$*
8. *Update $centroid(Cluster)$;*
9. *Assign the ID to transaction;*
10. *Count++*
11. *End if*
12. *Clusters(C_n) $\leftarrow cluster \cup newcluster$*
13. *End for*
14. *End for*

3.3 Heap Tree Formation:

A specialized tree refers heap tree based algorithm finds the patterns with maximum ID. A comparison based sorting algorithm divided the transaction set into two regions sorting and unsorted region. The iterative procedure of shrinking makes the unsorted region into sorted region with highest ID consider as root node. The heap tree construction algorithm receives the cluster of transactions as an input. Then, for each transaction in a cluster, it extracts the ID of transaction. The transaction array, transaction with maximum ID and the no. of transactions are considered for tree construction algorithm refers $todown()$. The tree construction process is repeated until all the clusters are used. In $todown()$ algorithm, the transaction with maximum ID is set it as root node. Then, the odd value of ID count is placed right to root node and the even values of ID count is placed at the left to root node. The insertion of new transaction to the heap tree consists of following processes:

Heap Tree

Input: *Clusters(C_n), count n*
Output: The tree T_H

1. *For each cluster C_i* for (*i=1….n*)
2. *for each transaction in cluster*
3. *Extract maximum ID*
4. *Start=maximum ID;*
5. *While(start≥0)*
6. *To down (Tr, start, count-1)*
7. *Start=start-1;*
8. *End for*
9. *End for*

- Check whether the new transaction is the last part of cluster
- Newly added transaction consider as the child node to the root node only if the condition is satisfied.
- Check the child node whether it is greater than or less than the root node.
- The child node will become the root node if it is greater than root node. Otherwise, it remains as it is.
- Finally, the heap tree is formed for all the transactions in the clusters.

To down(Tr, start, end)

1. *Root=start;*
2. *While root*2+1≤end do*
3. *Child=root*2+1;*
4. *Swap=root*
5. *If(Tr[swap]<Tr[child])*
6. *Swap=child*
7. *If child+1 ≤ end and Tr[swap]<Tr[child+1]*
8. *Swap=child+1*
9. *If Swap=root*
10. *Return*
11. *Else*
12. *Swap(Tr[root],Tr[swap])*
13. *Root=swap;*

The heap tree formation using an algorithm is explicitly implemented in MATLAB environment shown in Fig.2. From the Fig. 2, it is observed that the node 11689 consider as a root node. The first level of child are attached to left and right leaf nodes of root node 11689. Then, the sequential formation such a way that the nodes with even number of transaction attached left to the root node and with odd values attached to the right to the root node. Finally, the constructed tree by using the heap tree algorithm passed to the Space Partition algorithm.

3.4 Tree based space partition search:

The tree formation provided the necessary separation of item sets with maximum ID, even and odd. The tree structure contains n number of item sets. For each item set, frequent pattern extraction is provided by using the algorithm as follows:

Tree based space partition search

Input: *Item set in tree (n), transaction ID (T_{id})*
Output: *output data base (T)*

```
For k=1 to n
 For j=1 to i
If (j=1)
t[k, j]=1;
Else
t[k,j]=0;
End if
T[1, j]=t[T_id, j]VSP(T[1,j])
For each transaction in input file
For each item in transaction
Pos=(item-1)/log(n)
If (item%(2*n)=0)
Item =2*n;
Else
Item=item%2*n;
VSP[T_id, pos];
End if
End for
End for
```

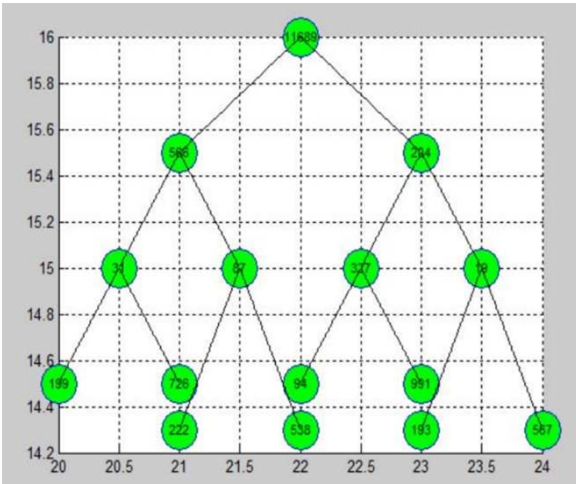

Fig. 2: Heap tree formation.

The item sets in the tree and transaction ID are given as the input to the algorithm. For each item set Vague Space Partition (VSP) of id and position are calculated. The position value is updated by using the logarithmic of ‘n’ number of transaction. The new position value is passed through the arguments of VSP which provided the frequent item set

3.4Space Partition:

Space Partition (SP) represents the frequent items provided the required compression and compaction of data. The item set in each transaction referred as frequent state if support value greater than the user defined threshold value. The distance between vague grid cells and trajectory locations are large. But, they are required to implement the trajectory transformation. The cells with the low membership values introduced the noise in transformed data. To eliminate noise, we set the threshold value (α) in the formation of membership function. The cells with membership values less than the threshold value (α) are filtered during the transformation process. The large threshold values damages the spatial approximation semantic structure. Hence, the ranges for membership threshold lies in between 15 % to 35 %. The input tree which represents the item sets in transactions transformed to the numerical data in the SP process. The transaction file represented by SP algorithm converted to array for future processing.

Vague Space Partition (VSP)

Input: *Tree data base (T), membership threshold(α), neighborhood value(n), length of data base(l),*

Output: *VSD*

1. *For each value in T*
2. *For each location point*
3. *Calculate the distance between two nodes*
4. *Choose the nearest neighbourhood and corresponding membership*
5. *Compare it with the membership threshold*
6. $VSD = items\ with\ minimum\ mebership\ value$
7. *End*
8. *End*
9. *Prefixspan(VSD)*

The transactions from the tree converted to item set format which is masked by heap tree architecture. The nearest neighborhood from space partition and the pre order transaction are performed. The comparison between the obtained results with the item sets in a membership threshold values form the mining patterns. Otherwise, the comparison between the transactions continued and the support level is calculated.The proposed VSP based on prefix span algorithm is proposed contains scanning process. The length of VSD obtained from VSP is calculated.

Prefix span

Input: *VSD, trajectory pattern (t), length (g), Projected database (Proj(VSD))*

Output: *Frequent pattern set*

1. *If (g=0)* *// Scanning process*
2. *Find frequent grid cell in projected database (Pro(VSD))*
3. *For each transaction (x) in VSD*
4. *t=x;*
5. *Record information* $\in (T_{id},\ time_{end}, support)$
6. *Else*
7. *Compare the time intervals for two sets* $(time_i, time_j)$
8. *Calculate support and append to information pair.*
9. *Accumulate total support in Proj(VSD).*
10. *End*
11. *Frequent pattern set=Information pair(* T_{id}, $time_{end}, support$*)*

Then, check whether the length is equal or greater than zero. If it is equal to zero, the information pair that contains transaction ID, time interval and support value are calculated. For length greater than zero, the new information pairs are calculated for each transaction. The time interval between the selected and neighborhood transaction are estimated. The supports corresponding to time are also calculated. The value of projected database updated with the new calculated support. The obtained information pair denoted the necessary frequent patterns. The vague space partition in frequent mining of item sets as shown in Fig. 3.

Australian Journal of Basic and Applied Sciences, 10(2) Special 2016, Pages: 250-261

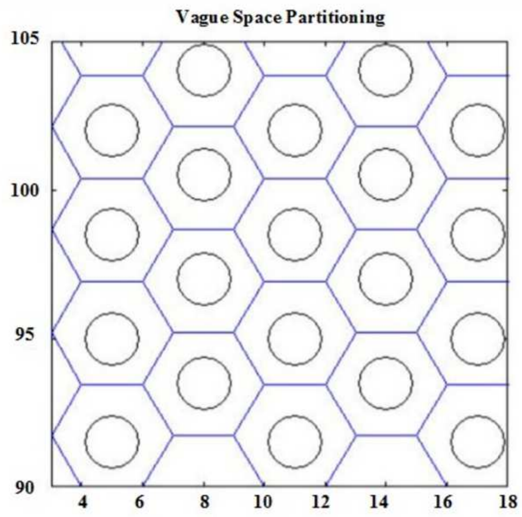

Fig. 3: Space partition.

RESULT AND DISCUSSION

The Tree based Space Partition of Trajectory Pattern Mining (TSPTPM) proposed in this paper used to obtain the frequent patterns in real time datasets available in Frequent Mining Item (FMI) set. The mining time analysis of proposed algorithm with various number of items is presented. The implementation of TSPTPM algorithm is carried out in Intel (R) Pentium (R) CPU G 630 (2.70 GHz) with 2 GB RAM memory with Windows 7. The algorithms are coded in MATLAB R2009b. The real time databases from http://fimi.ua.ac.be/data/ are used for implementation. The items required for cluster formation from the selected datasets are 500, 1000, 1500, 2000, 2500 and 3000 respectively. The evaluation of time with the items proves the effectiveness of TSPTPM frequent item sets mining.

Mining Time Analysis:

The analysis of mining time of proposed TSPTPM algorithm with the different cluster items are listed in Table 2.

Table 2: Mining time analysis

Cluster items	Mining Time Analysis							
	Chess		Mushroom		Connect		Accident	
	Apriori	TSPTPM	Apriori	TSPTPM	Apriori	TSPTPM	Apriori	TSPTPM
500	1.3196	0.8738	0.9403	0.6886	2.4182	0.8799	2.3036	1.2934
1000	1.3322	0.6301	0.9521	0.734	2.4475	0.941	1.3396	0.9956
1500	1.1039	0.629	0.9396	0.7776	2.5103	1.1073	1.3478	1.1599
2000	0.9933	0.6332	0.9207	0.841	2.5774	1.3119	1.3346	1.1211
2500	0.7871	0.6386	0.9333	0.8832	0.9333	0.8832	1.3247	1.101
3000	0.7314	0.6152	0.9636	0.8546	0.9636	0.8846	1.4008	0.956

The root node with maximum ID selection improved the space partition process. The analysis with various cluster items for all the data sets also presented in Fig. 4 to 7 respectively.

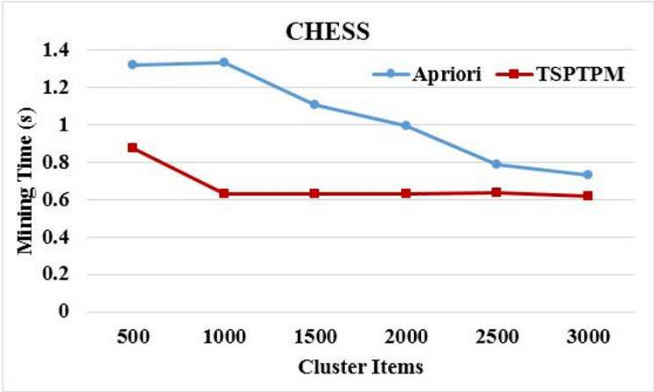

Fig. 4: Mining time analysis of two algorithms for chess dataset.

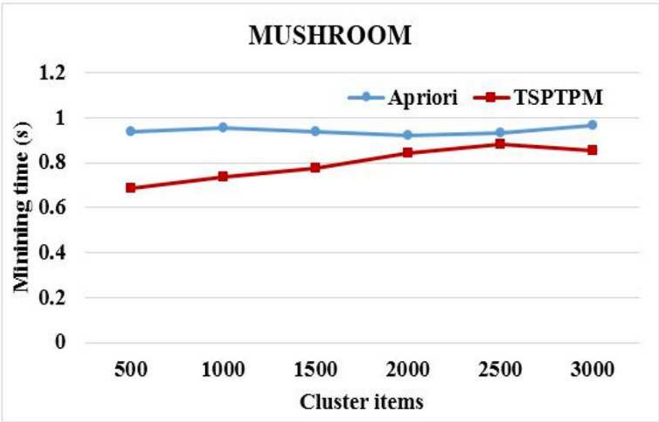

Fig. 5: Mining time analysis of two algorithms for mushroom dataset.

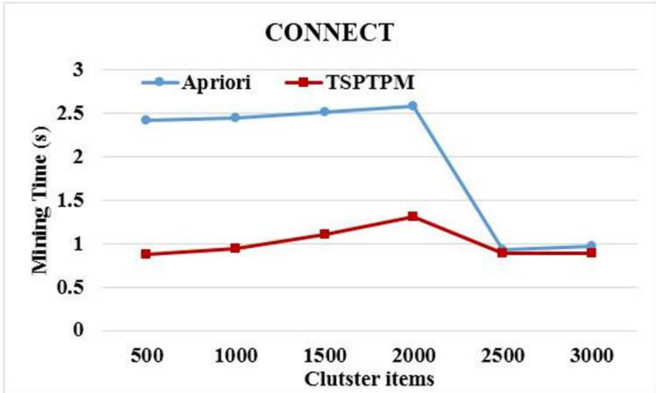

Fig. 6: Mining time analysis of two algorithms for connect dataset.

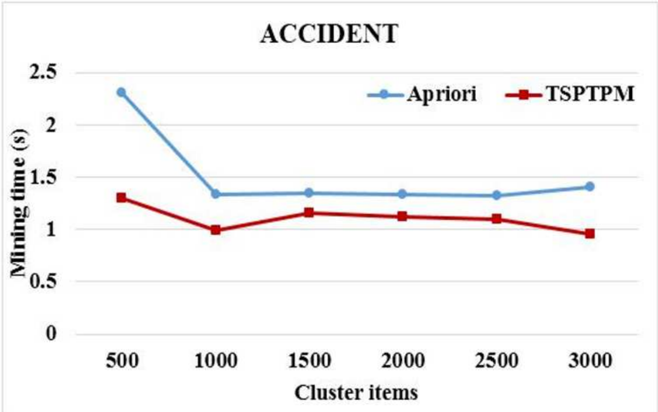

Fig. 7: Mining time analysis of two algorithms for accident dataset.

From the Fig.4 to 7, it is observed that for grouping of many items in the cluster, the value of mining time for TSPTPM is less compared to Apriori. The simplification provided by the tree based network optimized the partition approaches which minimizes the time consumption.

Conclusion:

Tree based Space Partition of Trajectory Pattern Mining (TSPTPM) algorithm is proposed in this paper in order to provide the reduction in time consumption. A novel method, which adopted clustering and tree structure for the extraction of the frequent patterns from the real time datasets provided by open source data mining library refers SPMF and National Institute of Statistics for the region of Flanders. Initially, the clustering process grouped the transactions and assigned the ID to each group. Then, heap tree algorithm constructed tree from the cluster of transactions. The transaction with maximum ID considered as a root node and the child nodes placed at left and right to a root node denoted odd and even number of transactions. Finally, Vague Space Partition (VSP) approach representation of frequent item started at root node provided the maximization in the density of sequences and assurance of effective data compression. The mining time analysis between the proposed TSPTPM and the existing Apriori for various number of items in each cluster proved the effectiveness in tree based frequent mining.

REFERENCES

Toon Calders, Nele Dexters, Joris J.M. Gillis, Bart Goethals, 2014. “Mining frequent itemsets in a stream”, Information Systems, pp: 233-255.

Bay Vo, Tzung-Pei Hong, Bac Le, 2011. “Dynamic bit vectors: An efficient approach for mining frequent itemsets”, Scientific Research and Essays, pp: 5358-5368.

Bay Vo, Tzung-Pei Hong, Bac Le, 2012. “DBV-Miner: A Dynamic Bit-Vector approach for fast mining frequent closed itemsets”, Expert Systems with Applications, pp: 7196-7206.

Bay Vo, Minh-Thai Tran, Hieu Nguyen, Tzung-Pei Hong,Bac Le, 2012. “A Dynamic Bit-vector Approach for Efficiently Mining Inter-sequence Patterns”, Third International Conference on Innovations in Bio-Inspired Computing and Applications (IBICA), 51-56.

Bay Vo, Tzung-Pei Hong, Bac Le, 2012. “Mining most generalization association rules based on frequent closed itemset”, International Journal of Innovative Computing Information and Control, pp: 7117-7132.

Bay Vo, Tzung-Pei Hong, Bac Le, 2013. “A lattice-based approach for mining most generalization association rules”, Knowledge-Based Systems, pp: 20-30.

Bay Vo, Tuong Le, Tzung-Pei Hong, Bac Le, 2014. “Maintenance of a frequent-itemset lattice based on pre-large concept”, Knowledge and Systems Engineering, Springer, pp: 295-305.

Basura Fernando, Elisa Fromont, Tinne Tuytelaars, 2012. “Effective use of frequent itemset mining for image classification”, Computer Vision–ECCV, Springer, pp: 214-227.

Liang Wang, David Wai-Lok Cheung, Reynold Cheng, Sau Dan Lee, Xuan S. Yang, 2012. “Efficient mining of frequent item sets on large uncertain databases”, IEEE Transactions on Knowledge and Data Engineering, pp: 2170-2183.

Chuntao Jiang, Frans Coenen, Michele Zito, 2013. “A survey of frequent subgraph mining algorithms”, The Knowledge Engineering Review, pp: 75-105.

Anthony J.T. Lee, Yi-An Chen, Weng-Chong Ip, 2009. “Mining frequent trajectory patterns in spatial–temporal databases”, Information Sciences, pp: 2218-2231.

Josh Jia-Ching Ying, Wang-Chien Lee, Tz-Chiao Weng, Vincent S. Tseng, 2011. “Semantic trajectory mining for location prediction”, Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp 34-43.

Weina Ge, Robert T. Collins, R. Barry Ruback, 2012. “Vision-based analysis of small groups in pedestrian crowds”, IEEE Transactions on Pattern Analysis and Machine Intelligence, pp: 1003-1016.

Yunhao Liu, Lei Chen, Jian Pei, Qiuxia Chen, Yiyang Zhao, 2012. “Mining frequent trajectory patterns for activity monitoring using radio frequency tag arrays”, IEEE Transactions on Parallel and Distributed Systems, pp: 2138-2149.

Liang Wang, Kunyuan Hu, Tao Ku, Xiaohui Yan, 2013. “Mining frequent trajectory pattern based on vague space partition”, Knowledge-Based Systems, pp: 100-111.

Arthur A. Shaw, N.P. Gopalan, 2014. “Finding frequent trajectories by clustering and sequential pattern mining”, Journal of Traffic and Transportation Engineering, pp: 393-403.

Kanad Basu, Prabhat Mishra, 2010. “Test data compression using efficient bitmask and dictionary selection methods”, IEEE Transactions onVery Large Scale Integration (VLSI) Systems, pp: 1277-1286.

Venkatesan, N., E. Ramaraj, 2011. “High Performance Bit Search Mining Technique”, International Journal of Computer Applications, pp: 15-21.

Sheila A. Abaya, 2012. “Association Rule Mining based on Apriori algorithm in minimizing candidate generation”, International Journal of Scientific & Engineering Research, pp: 1-4.

Jayshree Boaddh, Urmila Mahor, Niket Bhargava, 2012. “Empirical Evaluation of Bit Mask Search for Mining Frequent Item Sets”, International Journal of Engineering and Innovative Technology (IJEIT), pp: 1-15.

Jayshree Boaddh, Urmila Mahor, Niket Bhargava, 2012. “A Theoretical Formulation of Bit Mask Search Mining Technique for mining Frequent Itemsets”, International Journal of Emerging Technology and Advanced Engineering, 2: 12.

Robert D. Cameron, Kenneth S. Herdy, Dan Lin, Meng Lin, Ben Hull, Thomas S. Shermer, Arrvindh Shriraman, 2014. “Bitwise data parallelism in regular expression matching”, Proceedings of the 23rd international conference on Parallel architectures and compilation, pp: 139-150.

Geetha, P., E. Ramaraj, 2014. “A Frequent Trajectory Path Mining Using Bit Mask Search and UP Growth+ Algorithm”, World Congress on Computing and Communication Technologies (WCCCT), pp: 17-20.

Zhang Zhuo, Lu Yu-fan, Shen Wen-feng, Xu Wei-min, Zheng Yan-heng, 2011. “Blocking optimized SIMD tree search on modern processors”, Journal of Shanghai University (English Edition), pp: 437-444.

Junfu Yin, Zhigang Zheng, Longbing Cao, 2012. “USpan: an efficient algorithm for mining high utility sequential patterns”, Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, pp: 660-668.

Eng-Jon Ong Helen Cooper Nicolas Pugeault Richard Bowden, 2012. “Sign language recognition using sequential pattern trees”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp: 2200-2207.

Bay Vo, Frans Coenen, Bac Le, 2013. “A new method for mining Frequent Weighted Itemsets based on WIT-trees”, Expert Systems with Applications, pp: 1256-1264.

Gwangbum Pyun, Unil Yun, Keun Ho Ryu, 2014. “Efficient frequent pattern mining based on linear prefix tree”, Knowledge-Based Systems, 55: 125-139.

Unil Yun, Heungmo Ryang, Keun Ho Ryu, 2014. “High utility itemset mining with techniques for reducing overestimated utilities and pruning candidates”, Expert Systems with Applications pp: 3861-3878.