

# AN INTELLIGENT APPROACH FOR MINING FREQUENT SPATIAL OBJECTS IN GEOGRAPHIC INFORMATION SYSTEM

Animesh Tripathy<sup>1</sup> and Prashanta Kumar Patra<sup>2</sup>

<sup>1</sup> Department of Computer Science Engineering, KIIT University, Bhubaneswar, INDIA

<sup>2</sup> Department of Computer Science Engineering, CET, BPUT, Bhubaneswar, INDIA

## ABSTRACT

*Spatial Data Mining is based on correlation of spatial objects in space. Mining frequent pattern from spatial databases systems has always remained a challenge for researchers. In the light of the first law of geography “everything is related to everything else but nearby things is more related than distant things” suggests that values taken from samples of spatial data near to each other tend to be more similar than those taken farther apart. This tendency is termed spatial autocorrelation or spatial dependence. It’s natural that most spatial data are not independent, they have high autocorrelation. In this paper, we propose an enhancement of existing mining algorithm for efficiently mining frequent patterns for spatial objects occurring in space such as a city is located near a river. The frequency of each spatial object in relation to other object tends to determine multiple occurrence of the same object. We further enhance the proposed approach by using a numerical method. This method uses a tree structure based methodology for mining frequent patterns considering the frequency of each object stored at each node of the tree. Experimental results suggest significant improvement in finding valid frequent patterns over existing methods.*

## KEYWORDS

*Frequent Pattern, Ordered List, Spatial Data, Share Frequent Pattern, Conditional Pattern Tree*

## 1. INTRODUCTION

Geographic Information System stores objects in space known as spatial data in Spatial Database Systems (SDBS). SDBS are large database systems for the management of spatial data. To store and retrieve spatial data in order to find implicit regularities, relationships and autocorrelation rules hidden in the data has therefore become a challenge. It is important to analyze and develop different hypothesis from the spatial database. In contrast to mining in relational databases, spatial data mining algorithms have to consider the neighbours of objects in order to extract useful knowledge. This is necessary because the attributes of neighbouring object of interest may have a significant influence on the object itself. Frequent pattern mining [4] plays an essential role in many data mining tasks and applications, such as mining association rules [1], correlations [5], sequential patterns [2], classification [7] and clustering [2]. Methods for efficient mining of frequent patterns have been studied extensively by many researchers.

The main issues in frequent patterns mining are: to reduce the number of database scans since spatial transactional database is too large to fit into the main memory, and scanning data from disk is very costly; to reduce the search space since every subset of may be frequent pattern [1]. The main drawback is the generation of many frequent patterns. For example, if the pattern length is  $x$ , then  $2^x$  frequent patterns would be generated. In many applications with long

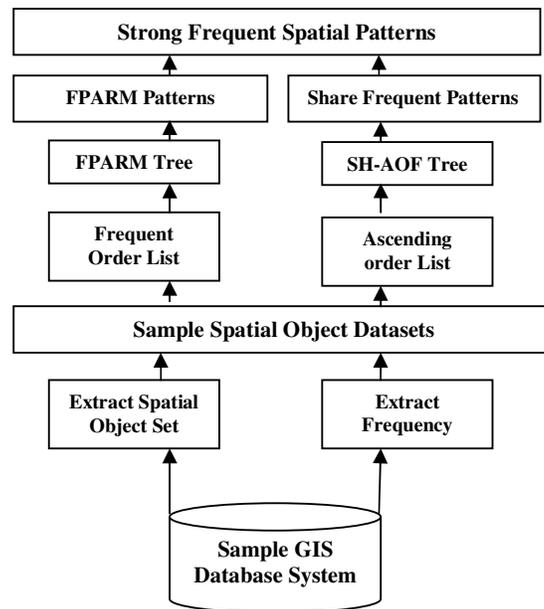
frequent patterns, enumerating all possible subsets is computationally not possible [8] in case of spatial data sets that usually contains long frequent patterns. The proposed approach differs mainly in how they represent the conditional databases that use a compact data structure [6] FP tree to represent the conditional databases, which is a combination of prefix-tree structure and node-links. In this paper, we propose an enhancement of existing FP tree by building a Frequent Pattern Association Rule Mining (FPARM) tree that has the capability of generating frequent spatial patterns. Many a time it is also necessary to record the frequency of occurrence of spatial objects and use the same in generation of frequent patterns. Secondly, we use a Share Ascending Order Frequent (SH-AOF) tree to efficiently mine the share information of each spatial object in a tree for finding frequent spatial patterns.

## 2. RELATIVE WORK DONE

Extensive efforts have been devoted to developing efficient algorithms for mining frequent patterns. A couple of algorithms adopting the candidate generate-and-test approach were proposed. Apriori algorithm [2, 3] is the first algorithm which uses the Apriori property to prune the search space. DIC algorithm [8] aims to reduce database scans. It starts counting support for an item set shortly after all of its subsets have been determined to be frequent rather than wait till the next database scan. A hash based algorithm [10] was proposed to reduce the number of candidate patterns. The pattern growth approach such as FP Tree [4, 11] tries to avoid candidate sets generation by constructing conditional databases for frequent patterns. It mainly represents a compact data structure, which is a combination of prefix-tree structure and node-links. Tree structures have been proposed to calculate all the frequent patterns using a single pass of database such as CP-tree [2] etc. This traditional frequent pattern mining problem considers only the binary occurrence (0/1), i.e. either absent or present of the items in one transaction. Share-confidence model to discover useful knowledge about numerical attributes associated with items in a transaction. ZP and ZSP [3] algorithms use heuristic methods to generate all the candidate patterns. Moreover, they cannot rely on the *downward closure* property as their searching method is very time-consuming and does not work efficient in large databases. Some other algorithms such as SIP, CAC and IAB [3] have been proposed to mine share-frequent patterns but they may not discover all the share-frequent patterns. Fast share measure (ShFSM), DCG [4, 6] improves the previous algorithms by using the *level closure* property. This property cannot maintain the *downward closure* property and wastes computation time in each pass. In this paper, we propose a novel tree structure to generates a very few candidates using a share measure and top-down traversal strategy with ascending frequency order which achieves significant performance improvement over previous works.

## 3. PROPOSED WORK

In this section we propose an approach to mine frequent patterns of spatial data in geographic information system. These spatial objects situated close to each other for a given sample space of Indian Cities. The presence of each spatial object is recorded against each city as a transaction. We form spatial objects data set for each transaction list of cities based of their occurrence of higher ranked objects. Equal ranked objects are sorted in lexicographic ordering [9,10]. The layout of the proposed model is given in figure 1. The Proposed Model can be described as a sequence of processes. We extract the spatial objects and its frequency of occurrences from the Geographic Map Database and build a Sample Spatial Objects Datasets. The next process is to first apply techniques to generate the FPARM Tree after getting the frequent order list. From the FPARM Tree we get the FPARM Patterns which we use in the final process to obtain the strong frequent spatial patterns. Similarly we again build an ascending order list and generate a SH-AOF Tree which gives us the share frequent patterns. The output of both FPARM Tree and SH-AOF Tree finally gives us the set of strong spatial frequent patterns.



**Figure 1.** Proposed Model

### 3.1. Mining Approach

In this section, we first introduce the concept of share frequent objects, and then describe an approach to find the frequent patterns from the transaction table. To find out the frequent patterns we consider a sample of 12 Indian Cities from the database table of 150 Indian cities. The 07 spatial objects such as Museum (A), Zoo (B), Lake (C), Monument (D), River (E), Forest (F) and Hill (G) are considered for analysis within a aerial distance of 15 square kilometres. We first find the Frequent Positive Association Rule Mining (FPARM) tree and then improve the generated tree and call it as Share-Ascending Order Frequent (SH-AOF) Tree. Initially we outline the mining approach to be a seven step process.

*Step 1: Scan Database to find frequent spatial objects for each city*

*Step 2: Build a FPARM tree by finding the Ordered list of objects.*

*Step 3: Generate patterns and validate it against their respective support count.*

*Step 4: Record frequency of each spatial object against every transaction*

*Step 5: Arrange each spatial object for every transaction in ascending order*

*Step 6: Calculate the Total frequency count for every transaction*

*Step 7: Generate the SH-AOF Tree and find the Valid Share Frequent Patterns*

The first step is to generate Frequent Order List of spatial objects after finding all frequent objects based on occurrence of objects in descending order in each transaction. Any object that do not satisfy the minimum support count is then simply discarded. The second phase of the algorithm builds the FPARM-Tree. For each transaction the frequent object pattern of current transaction is inserted into the tree. Then for successive transaction if any object is not found as a node in the tree, it creates a new object node and assigns one as the frequency. Otherwise the frequency of child node adds one. The FPARM tree is mined by starting from each frequent length pattern leaf node (as an initial suffix pattern), construct its pattern base (a “sub dataset,” which consists of the set of *prefix paths* in the FPARM-tree co-occurring with the suffix pattern), then construct its (*pattern*), and perform mining recursively on such a tree. The pattern growth is achieved by the concatenation of the suffix pattern with the valid frequent patterns generated from a valid FPARM-tree pattern in step three. The fourth step records the frequency of occurrence of each spatial object in the given city. The fifth step finds the sum of the

frequency count of each spatial object in all 12 transactions. In the same step we arrange the spatial objects in each transaction in ascending order. The sixth step is very necessary for generating the ascending order tree which finally generates valid share patterns in the last step.

In practice, considering the binary frequency ( either absent or present) or support of a pattern may not be a sufficient indicator for finding meaningful patterns of a transaction database because it only reflects the number of transactions in the database which contain that pattern.

Spatial autocorrelation is the tendency of spatial objects to get related due to their neighborhood ranking [5]. We can consider an example. Suppose, there are 3 museum, 4 monuments and 2 lakes are located a city (say New Delhi) and another city has 1 forest, 2 hills and 1 river. Therefore, traditional frequency/support measure cannot analyze the exact number of objects located in a city or can't discovering relation between objects in space. For that reason, to discover more important knowledge in association rule mining in spatial database system a 'Share Measure' concept was introduced which provided useful knowledge about the numerical values that are typically associated with the transactions objects. To overcome the "too many patterns" disadvantage we propose a method to discover frequent patterns. Let's now discuss the concept of frequent pattern first then share frequent pattern.

### 3.2. Frequent Pattern Mining

Let  $I = \{i_1, i_2, \dots, i_m\}$  be a set of items and  $D$  be a transaction database.  $\{T_1, T_2, \dots, T_n\}$  be a set of transaction where each transaction  $T_i \in D$  is a subset of  $I$ . The support/frequency of a pattern  $X\{x_1, x_2, \dots, x_p\}$  is the number of transactions containing the pattern in the transaction database. The problem of frequent pattern mining is to find the complete set of patterns satisfying a minimum support in the transaction database. The downward closure property (Agrawal et al., 1993; Agrawal and Srikant, 1994) is used to prune the infrequent patterns. This property tells that if a pattern is infrequent then all of its super patterns must be infrequent. Apriori (Agrawal et al., 1993; Agrawal and Srikant, 1994) algorithm is the initial solution of frequent pattern mining problem and very useful in association rule mining. But it suffers from the candidate generation-and-test problem and needs several database scans.

### 3.3. Share Frequent Pattern Mining

We have adopted similar definitions presented in the previous works (Carter et al., 1997; Barber and Hamilton, 2000, 2001, 2003; Li et al., 2005).

**Lemma 1:** *Given a transaction database and support threshold  $min\ sup$ , let  $F$ -list  $\{x_1 \dots x_n\}$  be a list of frequent objects. The complete set of frequent patterns can be  $n$  subsets as follows: the  $k$ th subset ( $1 \leq k \leq n$ ) contains patterns as  $(x_1, x_2, \dots, x_k)$ .*

Given a data set of spatial objects in each transaction, a pattern can be described as a subset that consists of items in the transaction.

**Lemma 2:** *The frequency count of each spatial object in a transaction is given as  $\{X_n\}$  where  $X$  belongs to the set of spatial objects and  $n$  denotes the occurrence of the object in the given transaction.*

As we have already discussed each spatial object can occur in a transaction more than one times. And we try to represent the same using a count for each object. The count is a numerical value greater than 1. This is known as share frequency of the each spatial object in the transaction.

**Lemma 3:** *The Count Set(CS) for each transaction against each occurrence of the spatial object is denoted as  $\{X_1, X_2, \dots, X_n\}$  where  $X_i$  represents a spatial object and  $n$  represents the frequency count for each.*

As per the discussion in this section and previous we need to keep track of the frequency count for every object in a given transaction and represent it using a set. The entries in the set represent the frequency count of each object.

**Lemma 4:** *The Total Frequency Count of a given transaction is denoted as  $SUM = \{X1_n+X2_n+...Xn_n\}$  where  $SUM$  is the summation of all frequency count of each object in a transaction.*

The total sum of each object’s frequency needs to be stored against each transaction. We do a simple summation of all frequencies in the transaction.

**Lemma 5:** *The Pattern Count (PC) for a pattern is given as summation of the occurrence of each pattern in every transaction.*

Every pattern may occur in each transaction. And so we need to store the total pattern count of each pattern as a sum of its individual count occurring in every transaction.

**Lemma 6:** *The Transaction Count of X is denoted by  $TC(X)$  as the total of  $SUM$  of X in every transactions occurring in the database.*

$TC$  expresses the total of  $SUM$  of each transaction where X is an object. A spatial database is generally a large data set of huge number of transactions. So  $TC$  is generally a large integer value as compared its corresponding  $PC$  value.

**Lemma 7:** *The Share Value (SV) for each pattern is represented as the ratio of Pattern Count (PC) and Total Database Count (TDC) where  $TDC$  represents the summation of  $TC$  values for each transaction of the database. It is represented as given below:*

$$SV(X)=PC(X)/ TDC$$

The share value actually represents the measure of the share frequent pattern in each spatial dataset. Based on the share value we find the validity of the pattern. If the  $SV$  is greater than the minimum threshold then it represents a valid pattern else invalid.

#### 4. ANALYSIS OF PROPOSED APPROACH

The proposed approach is used to mine strong frequent patterns. We have assumed a minimum support count greater than 4. Analysis process is described in detail as per the seven step approach described in previous section. Each transaction is scanned once to find frequent spatial object. It can be viewed as having in each transaction.

**Table 1.** Sample Spatial Objects Dataset

TID #	Reference City	Spatial Objects	Count	Frequent Order List	Count value after sorting
1	Bhubaneswar	A,C,D,B,E,G	2,3,4,1,1,1	A,B,C,D,E,G	2,1,3,4,1,1
2	Bangalore	A,C,D,B,F	2,2,2,1,3	A,B,C,D,F	2,1,2,2,3
3	Ajmer	A,C,D,E	1,2,3,1	A,C,D,E	1,2,3,1
4	Mumbai	A,C,D,B,E,G	1,3,4,2,1,2	A,B,C,D,E,G	1,2,3,4,1,2
5	Chandigarh	A,C,B,F	2,1,1,1	A,B,C,F	2,1,1,1
6	Trivandrum	A,C,B	3,4,1	A,B,C	3,1,4
7	Delhi	A,D,B,F,E	2,4,1,1,1	A,B,D,E,F	2,1,4,1,1
8	Ahmadabad	A,C,D,B,E	3,2,3,1,1	A,B,C,D,E	3,1,2,3,1
9	Pune	A,C,D,B,E,G	1,2,1,1,2,1	A,B,C,D,E,G	1,1,2,1,2,1
10	Mysore	C,D,B,F,G	3,2,1,1,1	B,C,D,F,G	1,3,2,1,1
11	Nagpur	A,C,D,B	2,3,4,1	A,B,C,D	2,1,3,4
12	Patna	A,D,B,E	1,1,1,4	A,B,D,E	1,1,1,4

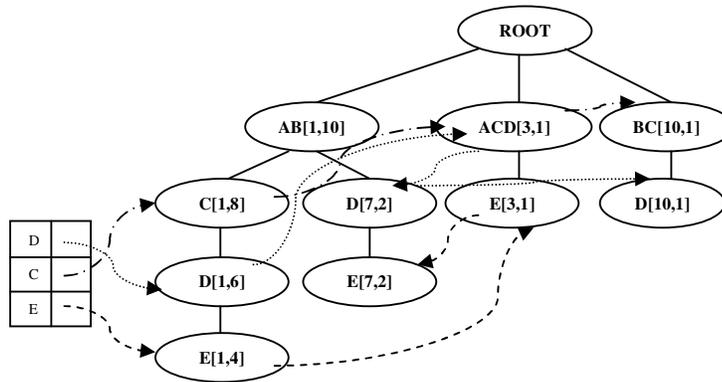


Figure 2. FPARM Tree

For example in TID (1) {A, C, D, B, E, G} is a transaction list. So the object which is less than the minimum support is pruned. Now TID (1) can be viewed as a new transaction {A, B, C, D, E}. Then, each transaction will be used to construct the FPARM tree as shown in figure. 2. Let’s take an example TID(1) for constructing the FPARM tree then Museum(A) and Zoo(B) can be combined into one node because these objects are more frequent than other in this transaction and so is inserted in to the child nodes of the sub tree. And similarly, other sub tree can be built and this procedure stops when no more transactions exists.

**4.1. Mining Frequent Patterns for Spatial Dataset**

Mining frequent patterns from the FPARM tree is summarized in Table 2. Suppose we consider leaf node C. C occurs in three branches of the FPARM tree of fig. 2. The paths formed by these branches are {A,B,C:8},{A,C:1} and {B,C:1}. Therefore, considering C as a suffix, its corresponding three prefix paths for generating patterns are {A, B: 8}, {A: 1} and {B: 1} which generate its patterns shown in Table 2. From the above we observe that the frequent patterns generated as shown above in the Table 2 do not take into account the occurrences of spatial objects having each frequency count greater than one and for mining each frequent pattern we have to search each branches of the tree which takes time. Therefore we again append the frequency of occurrence of spatial objects to the existing Table 1 and generate Table 3. Further taking this into account, we proceed our analysis with the construction of SH-AOF Tree.

Table 2. FPARM Pattern

Object	Generated Patterns	Valid Patterns	Valid Frequent Patterns Generated
C	{A,B:8},{A:1},{B:1}	{A:10,B:9}	{AB,C:8},{A,C:1},{B,C:1}
D	{A,B,C:6},{A,C:1}, {A,B:2},{BC:1}	{A:9,B:9,C:8}	{A,B,C,D:6},{A,C,D:1}, {A,B,D:2},{B,C,D:1}
E	{A,B,C,D:4},{A,B,D:2},{A ,C,D:1}	{A:7,B:6,C:5, D:7}	{A,B,C,D,E:4},{A,B,D,E:2},{A,C,D, E:1}

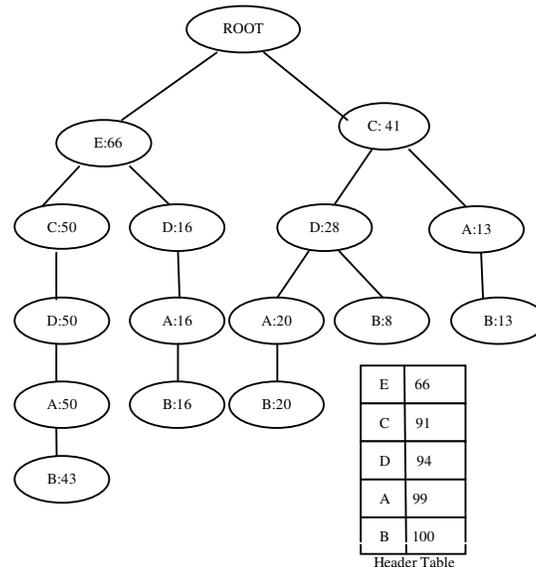
**4.2. SH-AOF Tree Construction**

In this section, we describe the construction process of the SH-AOF-Tree for frequent pattern mining. A Header table is maintained which contain object name and TC values of objects. In the first database scan, SH-AOF -Tree captures the TC value of all the items. Consider the database shown in Table 3 and minimum SV= 0.4.

**Table 3.** Spatial Database Transaction with count

TID #	Ordered Spatial Objects	Count	Total Count
1	E,C,D,A,B	1,3,4,2,1	12
2	C,D,A,B	2,2,2,1	10
3	E,C,D,A	1,2,3,1	7
4	E,C,D,A,B	1,3,4,1,2	13
5	C,A,B	1,2,1	5
6	C,A,B	4,3,1	8
7	E,D,A,B	1,4,2,1	9
8	E,C,D,A,B	1,2,3,3,1	10
9	E,C,D,A,B	2,2,1,1,1	8
10	C,D,B	3,2,1	8
11	C,D,A,B	3,4,2,1	10
12	E,D,A,B	4,1,1,1	7

According to lemma 5,  $\min PC = 43(\text{round}(0.4 * \text{TDC}))$ . After the first database scan, the TC values of the objects {A:99,B:100,C:91,D:94,E:66,F:32,G:41}. To be a candidate share-frequent object, the TC of an object must be at least 43. Therefore, the objects “F” and “G” are not candidate objects. According to the downward closure property, we can simply discard these objects. Next, we sort the header table in ascending order according to TC values of the objects. The header table order of objects is {E:66, C:91,D:94,A:99 and B:100}. For the first transaction #1 in Table 2, which contains objects {A, B, C, D, E, G}, we discard the non-candidate object G at first and then arrange the objects according to the header table sort order. Objects {E, C, D, A, B} (shown in Table 3) get the SUM value of #1 is 12. We build the SH-AOF Tree in a top down manner. The objects are inserted as the child of the root node. While inserting a new node, the TC value of the node is added with previous TC value. Figure1 shows the final SH-AOF-Tree with each spatial object’s TC value of transaction database. After the SH-AOF-Tree is constructed, the mining of share frequent pattern will be performed. No repeated scan of the original database is required. Next section performs mining operation in SH-AOF Tree.



**Figure 3.** SH-AOF Tree

### 4.3. Mining Process of SH-AOF Tree

The SH-AOF Tree is a compact representation of the conditional databases. It contains the complete information for mining frequent objects from the original database. In the SH-AOF Tree, the conditional database of an object (say x) is the union of the sub trees rooted at nodes with item x. For example, in Figure 2, the conditional database of object E is completely represented by the first sub tree of the root because the first child of the root is the only node with object E. The conditional database of object C contains two sub trees, one is the first sub tree of node E and the other one is the second sub tree of the root.

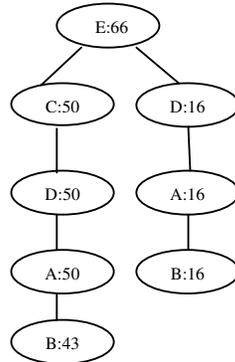


Figure 4. Conditional Pattern of E

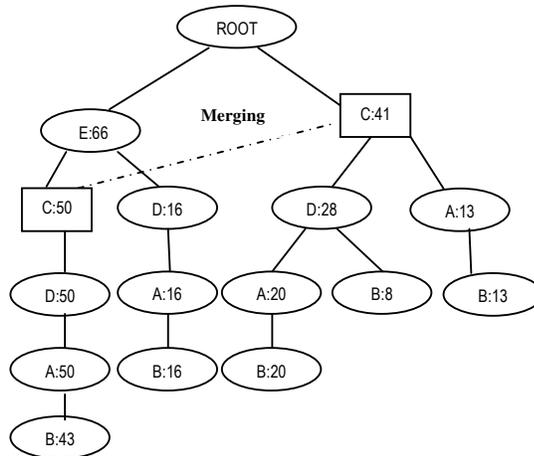
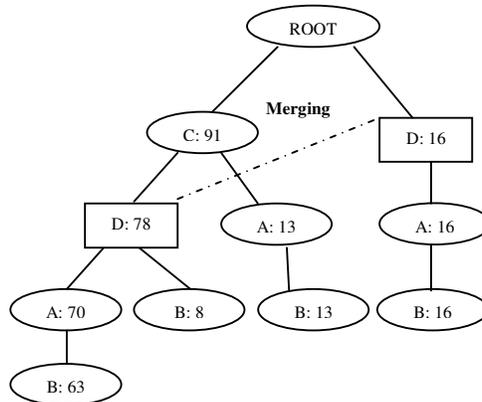
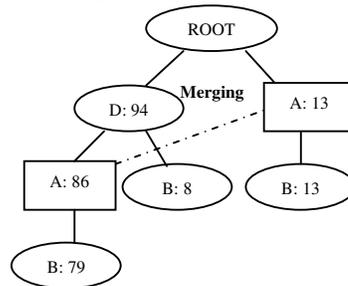


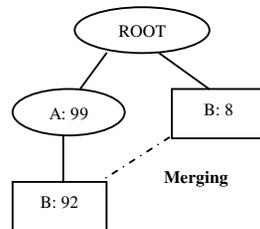
Figure 5. Merge to get conditional tree of C



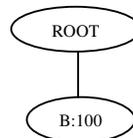
**Figure 6.** Merge to get conditional tree of D



**Figure 7.** Merge to get conditional tree of A



**Figure 8.** Merge to get conditional tree of B



**Figure 9.** Conditional Pattern of B

The SH-AOF Tree traverses the search space in depth-first order. It first mines the first conditional database E (the first sub tree of the root), then mine E's descendants conditional databases until all the patterns with E as prefix are mined. After the mining on E is finished, E will not be needed in future mining. All of its siblings will be merged with existing sub trees. The merging operation involves mainly pointer adjustment and support update. It can be done by traversing the two trees at the same time. During the traversal it accumulates the supports of the same nodes and adjusts pointers if two same nodes have the same set of children. A new SH-AOF is built from E as shown in Figure 2 which represents the conditional databases of E. After merging operation, the second sub tree of the root i.e. C becomes the first sub tree of the root, and it is the complete representation of the second object's conditional database. Then mining will be performed on C. The candidate patterns are generated from the object which satisfies minimum TC value and the final share frequent pattern is created from those candidate patterns which satisfy minimum share value. So the conditional patterns of C are {CA, CB, CD, CAB, CDA, CDB, CDAB} as the TC values of these patterns is greater than minimum PC value of 43. But only {CAB, CDA, CDB, CDAB} patterns have SV values greater than 0.4. So these are share frequent patterns.

**Table 4.** Share frequent patterns

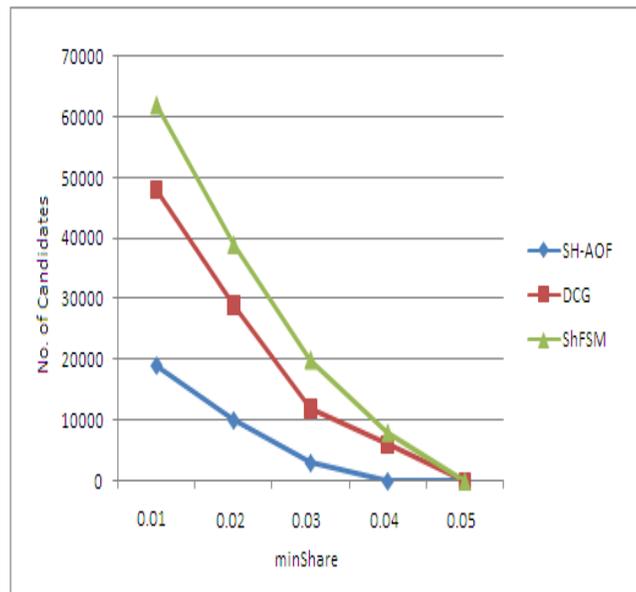
Pattern No	Frequent Patterns	SUM	PC	SV-Value
1	CDA	70	50	0.467
2	CDB	71	46	0.429
3	CDAB	63	51	0.476
4	CAB	76	45	0.420
5	DAB	79	46	0.429

#### 4.4. Generation of Strong Frequent Spatial Patterns

Now let us summarize the complete generation process of frequent spatial patterns. Performing mining on other sub trees of SH-AOF Tree, we generate the complete set of share frequent patterns such as {CAB, CDA, CDB, CDAB, DAB} that satisfy the minimum threshold value. In section 4.1 we had obtained the frequent spatial patterns such as {ABC, AC, BC, ABCD, ACD, ABD, BCD, ABCDE, ABDE, ACDE} using FPARM Tree approach. FPARM approach improved the process of generating strong frequent spatial patterns as compared to traditional algorithms while SH-AOF approach generated the share frequent spatial patterns taking into account the frequency of occurrence. This concept provided useful knowledge about the numerical values that are typically associated with the transactions objects. Therefore merging the frequent patterns generated in both FPARM and SH-AOF gives us the final set of strong frequent spatial patterns.

### 5. EXPERIMENTAL RESULTS

In this section, we report our experimental results on the performance of SH-AOF tree with respect to efficiency and scalability in comparison with DCG and ShFSM. It shows that our approach outperforms DCG and ShFSM and is efficient and highly scalable for mining very large databases. To evaluate the performance of our proposed tree structure, we have performed experiments on IBM synthetic dataset *T10I4D100K* from frequent item set mining dataset repository (<http://fimi.cs.helsinki.fi/data/>) and UCI Machine Learning Repository (<http://kdd.ics.uci.edu/>). All the experiments were performed on a 2.0 GHz Pentium PC machine with 512 MB main memory and 40 GB hard disk, running Microsoft Windows/NT. FPARM, SH-AOF, DCG and ShFSM were implemented by us using Visual C++6.0. These datasets provides binary quantity of each item for each transaction and we have generated random numbers for the quantity of each item in each transaction, ranging from 1 to 10. The performance of our algorithm is better than the existing algorithms in number of candidate comparisons as shown in Figure 10 for various spatial data sets, with consistent results. Limited by space, only the results on some typical data sets are reported here.



**Figure 10.** Comparison Results of SH-AOF with DCG and ShFSM.

## 6. CONCLUSION

The proposed approach will definitely enhance the efficiency of finding spatial objects in space with respect to their correlation analysis. The experiments verified the compactness of the data transformation technique. The generation of frequent patterns is done by using FPARM and SH-AOF approach reduces the search space as well as gives more information of correlation of spatial objects. SH-AOF tree gives share information of each spatial object. Our approach prunes huge number of unnecessary candidates during tree creation time. It eliminates non-candidate single-element patterns which are sorted in ascending frequency order. During mining time it traverses the prefix-trees in top-down depth-first order. The combination of these two methods is more efficient than existing approaches. Moreover, FPARM and SH-AOF approach is very simple and gives more information about the strong frequent patterns. The combination of these two methods is more efficient than existing DCG and ShFSM approaches. Extensive performance analysis shows that our approach is more efficient and overcomes limitations of previous methods.

## REFERENCES

- [1] Xiong, H., Tan, P.-N. and Kumar, V. (2006), Hyperclique Pattern Discovery, *Data Mining and Knowledge Discovery*, Vol. 13, pp. 219-242.
- [2] Tanbeer, S. K., Ahmed, C. F., Jeong, B. -S. and Lee, Y. -K. (2008), CP-tree: A tree structure for single pass frequent pattern mining, in *Proceedings of the 12<sup>th</sup> Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD)*, pp. 1022-1027.
- [3] Barber, B. and Hamilton, H.J. (2003), Extracting share frequent itemsets with infrequent subsets, *Data Mining and Knowledge Discovery*, Vol. 7, pp. 153-185.
- [4] Li, Y.-C., Yeh, J.-S. and Chang, C.-C. (2005a), A fast algorithm for mining share-frequent itemsets, in *Proceedings of the 7th Asia-Pacific Web Conference on Web Technologies Research and Development (APWeb)*, Lecture Notes in Computer Science, Vol. 3399, Springer-Verlag, Berlin, pp. 417-428.

- [5] Jian Pei, Jiawei Han, Hongjun Lu, Shojiro Nishio, Shiwei Tang and Dongqing Yang (2007) H-Mine: Fast and space-preserving frequent pattern mining in large databases, *IEEE Transactions* (2007) , pp.593–605.
- [6] Li, Y.-C., Yeh, J.-S. and Chang, C.-C. (2005b), Direct candidates generation: a novel algorithm for discovering complete share-frequent itemsets, in *Proceedings of the 2nd Intl. Conf. on Fuzzy Systems and Knowledge Discovery (FSKD)*, Lecture Notes in Artificial Intelligence, Vol. 3614, Springer-Verlag, Berlin, pp. 551-560.
- [7] Wei, J. -M., Yi, W. -G. and Wang, M. -Y. (2006), Novel measurement for mining effective association rules, *Knowledge-Based Systems*, Vol. 19, pp. 739-743.
- [8] Liu, G., Tsai, Lu, H. and Yu, J. X. (2007), CFP- tree: A compact disk-based structure for storing and querying frequent itemsets', *Information Systems*, Vol. 32, pp. 295-319.
- [9] A.Tripathy, S.Das, P.K.Patra, An Improved Design Approach in Spatial Databases Using Frequent Association Rule Mining Algorithm, *2<sup>nd</sup> IEEE IACC 2010, IEEE Press*, pp 404-409.
- [10] A.Tripathy, S.Das, P.K.Patra, A Framework for Mining Strong Positive and Negative Association Rules of Frequent Objects in Spatial Database System Using Interesting Measure, *Recent Trends in Network Security and Applications Communications in Computer and Information Science, Springer Germany*, Volume 89, Part 3, pp 583-592, 2010.
- [11] Grahne, G. and Zhu, J. (2005), 'Fast Algorithms for frequent itemset mining using FP-Trees', *IEEE Transactions on Knowledge and Data Engineering*, Vol. 17, no. 10, pp. 1347-1362.