Incremental Mining of Frequently Correlated, Associated-Correlated and Independent Patterns Synchronously by Removing Null Transactions

Md. Rezaul Karim[1], Azam Hossain[1], A.T.M Golam Bari[1], Byeong-Soo Jeong[1], and Ho-Jin Choi[2]

[1]Department of Computer Engineering, Kyung Hee University, Korea
[2]Dept. of Computer Science, Korea Advanced Institute of Science and Technology, Korea
E-mail: [1]{asif_karim, azam, bari, jeong}@khu.ac.kr; [2]hojinc@kaist.ac.kr

Abstract- Frequent itemset mining leads to the discovery of associations and correlations among items in large transactional data sets. The discovery of interesting correlation relationships among huge amounts of business transaction records can help in many business decision-making processes, such as cross-marketing and customer shopping behavior analysis. Associated and correlated items are placed in the neighboring shelf to raise their purchasing probability in a super shop. Therefore, the mining combined association rules with correlation can discover frequently correlated, associated-correlated and independent patterns synchronously, that are extraordinarily useful for making everyday's business decisions. Since, the existing algorithms for mining correlated patterns did not consider the overhead of 'null transactions' during the mining operations; these algorithms fail to provide faster retrieval of useful patterns incrementally; besides, memory usages also increase exponentially. In this paper, we proposed an efficient algorithm namely 'IACAI' for mining above mentioned four kinds of patterns by removing so called 'null transactions'; by which not only possible to save precious computation time but also speeds up the overall mining process. Comprehensive experimental results show that the technique developed in this paper are feasible for mining large incremental transactional databases in terms of time, memory usages, and scalability.

Keywords: Associated patterns, correlated patterns, associated-correlated patterns, independent patterns**,** incremental transactional database, null transactions, market basket analysis.

1 Introduction

Data mining is defined as the process of discovering significant and potentially useful patterns in large volume of data. One objective of association rule mining is to discover correlation relationships among a set of items. One difficulty is how to select a proper interestingness measure that can effectively evaluate the associated degree of patterns, as there is still no universally accepted best measure for judging interesting patterns [6]. The well-known algorithm for finding association rules in large transaction databases is Apriori [12]. On the other hand, correlation mining is much more effective because of the large number of correlation relationships among various kinds of items. However, an independent pattern might have a much more probability than a correlated pattern to be a novel paired or grouped items even if they have the same support for the sake of the downward closure property of independence [1, 13]. In the recent market the concept of super shop is very popular among the peoples since, these shops keep almost everything according to customers preferences and very often these super shops has lots of branch around a country so, the number of transaction and purchase is huge; hence to predict e-shoppers or customer's purchase behavior changes with times. An organization's management first identifies target e-shoppers who share similar preferences for products and looks for those products that target e-shoppers are most likely to purchase. The purchase transactional records of e-shopper are used to build e-shoppers' profile describing his or her likes and dislikes. A set of e-shoppers known as neighbors who have exhibited

similar behavior in the past, can be found through calculating the correlations among e-shoppers [14].

In reality data changes from time to time in many areas, including the retail industry and the financial sector. Therefore, the itemsets mined in these wide applications can present some development trends. When the transaction database changes with time, dynamically increments, some new frequent itemsets can appear, and some old frequent itemsets can disappear, which induces the incremental mining. Therefore, in this paper, we proposed an efficient incremental mining approach for frequently correlated, associated-correlated and independent patterns synchronously by removing null transactions; which not only saves the mining time and memory usages but also speeds up the overall mining process.

The rest of this paper is organized as follows. Section 2, describes related works and the motivation behind this research. Section 3, represents the problem formulation. Section 4, represents our proposed approach and the 'IACAI' algorithm. In section 5 we devised some experimental results. Conclusions are presented in section 6. In this paper we used the term 'itemsets' and 'patterns'; 'database' and 'datasets' interchangeably.

2 Related Works and Motivations

2.1 Related Works

Many research works have been done in the field of correlated frequent pattern mining. Most of them first generate frequent itemset then uses these frequent itemset to mine correlated patterns. Including many of these algorithms are Apriori based [14, 15], for this reason is not scalable and is impractical for many real-time scenarios. FP-growth mining algorithm [9], offers better performance than Apriori algorithm as the former does not depend on candidate generation. Also, the database is fully scanned just twice. However, FP-tree algorithm does not drop the so called '***null transactions***' for subsequent scanning of conditional databases. Also, when the patterns are too long and redundant, it is impractical to construct a main- memory based FP-tree. Algorithms based on mining maximal frequent itemsets performs better than FP-tree based algorithms since, they avoid redundant patterns [16]. However, the maximal frequent itemset mining does not give complete information on the frequent itemsets, unlike algorithms based on closed frequent itemset mining. Also, they consider ***'null transactions'*** for mining, which is avoidable. The stream based algorithms uses FP-tree for representing all frequent itemsets which is obtained by scanning all transactions, including null transactions [14, 17, 18]. Transactions which contain just a single itemset can be avoided from the scheme of things even in stream data since; it cannot help in representing any pattern. Zhun et al. [19] have proposed a modified FP-tree which is built obviously by scanning every transaction including null transactions. This approach however requires all transactions to be considered for mining. On the other hand, Miccinski ***et al.*** [4], introduced three alternative interestingness measures, called any-confidence, all-confidence and bond for mining associations for the first times ever.

Later on, Y.K Lee ***et al***. [3, 11] used all-confidence to discover interesting patterns although both of them defined a pattern which satisfies the given minimum all-confidence as a correlated pattern. B. Liu et al. [2], used contingency tables for pruning and summarizing the discovered correlations etc. In this paper, a new interestingness measure corr-confidence is proposed for correlation mining. After that, **Z.** Zhou [1], mines all independent patterns and correlated patterns synchronously in order to get more information from the results by comparing independent patterns with correlated patterns. An effective algorithm is developed for discovering both independent patterns and correlated patterns synchronously, especially for finding long independent patterns by using the downward closure property of independence. In the literature [13], Z. Zhou, combines association with correlation in the mining process to discover both associated and correlated patterns. A new interesting measure corr-confidence is proposed for rationally evaluating the correlation relationships. This measure not only has proper bounds for effectively evaluating the correlation degree of patterns, but also is suitable for mining long patterns.

Actually mining long patterns is more important because a practical transactional database may contain a lot of unique items. However, these works built obviously by scanning every

transaction including null transactions and most of them were static mining approach and did not consider incremental mining approach.

2.2 Motivations and the Screening of the Null Transactions

A null transaction is a transaction that does not contain any item-sets being examined. Typically, the number of null-transactions can outweigh the number of individual purchases because, for example, many people may buy neither milk nor coffee, if these itemsets are assumed to be two of the frequent itemsets. So, it is highly desirable to have a measure that has a value that is independent of the number of null-transactions. A measure is null-invariant if its value is free from the influence of null-transactions [9]. From the previous section we observed that a lots of good works have been proposed and developed [1, 2, 4, 11, 13], but the performance degrades drastically especially when the transactional datasets are sparse due to the presence of null transactions.

Unfortunately, above mentioned works do not have the null-invariance property. Since, large data sets typically have many null-transactions, it is important to consider the null-invariance property when selecting appropriate interestingness measures for pattern evaluation. In this proposed approach an attempt has been made to eliminate the null transactions thereby, attempting to reduce the processing time for finding frequent *k*-itemsets. Finding null transactions and later eliminating them from future scheme of things is the initial part of this proposed framework. Consider for instance that, an electronic shop has 100 transactions of which, 40% are null transactions. FP-tree method of mining or any other related method in that case would scan all the 100 transactions while, our proposed approach attempts to reduce the transactions to 60% by considering just the valid 60 transactions after screening the 40 null transactions. This saves a lot of precious computation time 9]. Besides, an attempt has been made to find null transactions by using vertical data layout format [20]. It is quite possible to find the null transactions by finding those transactions that don't appear against any frequent single-itemset with this representation.

3 Problem Definition

In this section, we first define the problem of correlated, associated, associated- correlated and independent patterns mining and then present some preliminary knowledge that will be used in our algorithm adopted from literature [1, 13] *et al*. Suppose we have a transactional database DB in table 1 the problem is that mining the complete set of correlated, correlated-associated and independent pattern efficiently. In statistical theory, $A_1, A_2 ....A_n$ are independent if $\forall k$ and $\forall 1 \leq i_1 {}_2 \leq ....<i_k \leq n$,

$$P(Ai_1 Ai_2 .....Ai_k) = P(Ai_1) P(Ai_2) ...P(Ai_k) \qquad (1)$$

1. If a pattern has two items, such as pattern *AB*, then,

(2)

2. If a pattern has more than two items, such as pattern X= $\{i_1, i_2...i_n\}$, then

(3)

From (2) and (3), we can see that ρ has two bounds, i.e. $-1 \leq \rho \leq 1$. Let ô be a given minimum corr-confidence, if pattern *X* has two items *A, B and* if $|\rho(AB)| >$ ô, then *X* is called a correlated pattern or *A* and *B* are called correlated with each other, else *A* and *B* are called independent. If pattern *X* has more than two items, we define a correlated pattern and an independent pattern as follows:

Definition 1: Correlated pattern- Pattern *X* is called a correlated pattern, if and only if there exists a pattern *Y* which satisfies $Y \subseteq X$ and $|\rho(AB)| >$ ô; where ô is a predefined value of ρ.

Definition 2: Independent pattern- If pattern *X* is not a correlated pattern, then it is called an independent pattern. Now we define the associated patterns.

Let T= {i_1, i_2, i_3... i_m} be a set of m distinct literals called items and *D* is the set of variable length transaction over T. Each transaction contains a set of items, {i_{j1}, i_{j2}, i_{j3}... i_{jk}} ⊂T. Pattern X is a subset of T. In table 1, ***DB*** indicates the original dataset and db^+ indicates the incremental part of the dataset. The interestingness measure all-confidence denoted by α of a pattern X can be defined as follows:

$$a(X) = \frac{Sup(X)}{Max\_item\_Sup(X)} \quad (4)$$

Definition 3: Associated pattern- A pattern is called an associated pattern, if its all-confidence is greater than or equal to the given minimum all-confidence threshold.

Definition 4: Associated-correlated pattern- A pattern is called an associated-correlated pattern if it is not only an associated pattern but also a correlated pattern. Let pattern *X* be an associated-correlated pattern, then it must have two subsets *A* and *B* which satisfy the condition that the sale of *A* can increase the likelihood of the sale of *B*.

Example 1. For the filtered transactional database in Table 3, we have α (*AC*) =3/4 and α (*CE*) = 3/4. We also have,

$$\rho(AC) = P(AC) - P(A)P(C) / P(AC) + P(A)P(C) = 1/5 \text{ and}$$

$$\rho(CE) = P(CE) - P(C)\,P(E) / P(CE) + P(C)\,P(E) = 1/17$$

Let, the given minimum all-confidence set to be 0.35 and the given minimum corr-confidence set to be 0.10, then both *AC* and *CE* are associated patterns. However, pattern *AC* is a correlated pattern and pattern *CE* is an independent pattern. Therefore pattern *AC* is an associated-correlated pattern and pattern *CE* is an associated but not correlated pattern.

Table 1. An incremental transactional database

TID	Items	Part
10	A, B, C, F	DB
20	C, D, E	
30	A, C, D, E	
40	A	
50	D, E, G	
60	B, D	
70	B	
80	A, C, E	
90	F	
100	**A, C, D**	db^+
110	**G**	
120	**B, D, E**	

4 Proposed Approach

4.1 Work Flow of the Proposed Approach and the ACAI Algorithm

We mine all frequent correlated, associated, associated-correlated and independent patterns in two steps. First, we discover all frequent patterns using FP-growth [9], and then test whether they are correlated, associated, associated-correlated and independent patterns or not based on constraints defined by definition 1 through 4. For this we us two level pruning. For level 1 pruning we perform it by removing 'null transaction' and minimum support threshold. On the other hand level 2 pruning is performed by the constraints defined by definitions 1 through 4.

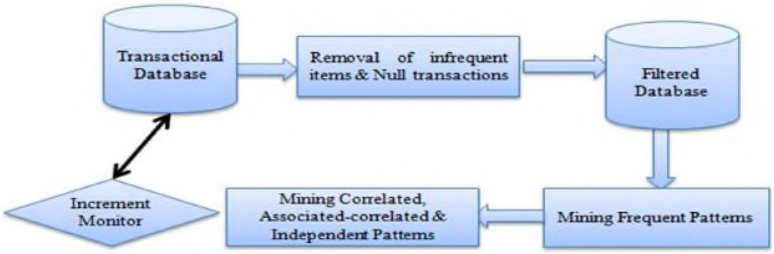

Fig. 1. Workflow of our proposed approach

The IACAI Algorithm

Input: i) An incremental transactional database ii) min_sup, iii) min_corr_conf, and iv) min_all_conf.
Output: Complete set of frequently correlated, associated, associated-correlated and independent patterns from that database.
Step-1: Scan the original dataset then remove 'null transactions' and 1-infrequent itemsets.
1. Represent the dataset using vertical layout format.
2. Find & remove null transactions from the dataset.
3. Periodically scan the dataset to see the increment of new transaction(s).

Step-2: Scan the filtered dataset and apply the modified FP-growth to generate the frequent itemsets. Suppose X is a frequent itemset.
1. Prune frequent 1-itemssets.
2. If X has 2 or more items and S(X) ≥ min_sup X is a frequent pattern. //Candidate patterns **Step-3:** Find the complete set of frequent correlated, associated, associated-correlated and independent patterns. Suppose X is a frequent pattern.
1. If (∂(X)>min_corr_conf)
   X is a correlated pattern
2. Else If (|(α(X))|>min_all_conf) X
   is an associated pattern.
3. Else if (∂(X)>min_corr_conf &&|(α(X))|>all_conf)
   X is an associated-correlated pattern
4. Else
   X is an independent pattern.

Fig. 2. The IACAI Algorithm.

4.2 An Example

Now, consider the vertical layout format representation of the same database given in table 2. Here, transaction 10 and 50 are supposed to be the '***null transactions***'; also from table 1 transactions 40, 70 and 90 are null transactions and are already not been considered for mining. In the incremental part TID 110 is also a null transaction. It is clear that, the null transactions containing just 1-itemsets are not significant since these itemsets do not have contribution while mining correlated patterns or association rule mining, hence, have been removed prior to mining. Also, itemsets F and G do not satisfy the minimum support count of 2, and is hence, avoided for mining.

For the ease of the reader we just showed the corresponding filtered transactional database in table 3 of the original database given by table 1. Let, the given minimum all-confidence is set to be 0.45 and the minimum corr-confidence is 0.10.The resultant FP-tree formed from the dataset given in table 3 is shown in fig. 1.

Table 2. Vertical layout format of the example database

Items	TID Sets
A	10, 30, 80, 90
B	60, 100
C	20, 30, 80, 90
D	20, 30, 50, 60, 90, 100
E	20, 30, 50, 80, 100
F	10
G	50

Since, our objective is to mine frequently correlated patterns family; hence intentionally we avoided the details of mining frequent patterns from the FP-tree. It is to be noted that, our proposed algorithm will not consider transactions 10, 40, 50 and 70 while scanning the dataset (Table 2) for the second time to construct the FP-tree since, they are null transactions.

The figure 2 shows the formal algorithm of our proposed work and fig 3 shows the support count of 1-itemsets and resultant FP-Tree. And table 4 shows the resultant frequent patterns with their corresponding supports. And we applied the constraints defined by definition 1 through 4 to mine the correlated, associated, associated-correlated and independent patterns and have shown in figure 4.

Table 3. The Filtered Database of the original database presented in table 1

TID	Items
20	C, D, E
30	A, C, D, E
60	B, D
80	A, C, E
100	A, C, D
120	B, D, E

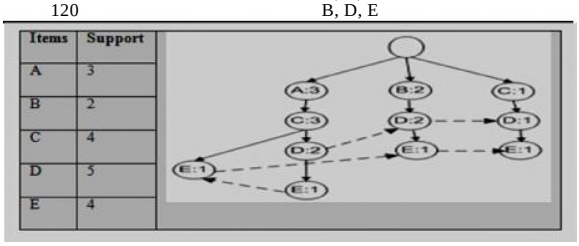

Fig. 3. Support count of 1-itemsets and resultant FP-Tree

Table 4. Frequent patterns and their support from the FP-Tree

Frequent Pattern	Support	Frequent Pattern	Support
AC	3	CE	3
AD	2	DE	3
AE	2	ACD	2
BD	2	ACE	2
CD	3	CDE	2

Frequent Pattern (X)	| ρ(X)|	α(X)	Correlated	Associated	Associated- Correlated	Independent
AC	0.2000	0.75	√	√	√	
AD	0.1200	0.40	√			
AE	0.0000	0.50		√		√
BD	0.0909	0.40				√
CD	0.0715	0.60		√		√
CE	0.0600	0.75		√		√
DE	0.0715	0.60		√		√
ACD	0.0435	0.40				√
ACE	0.2000	0.50	√	√	√	
CDE	0.0527	0.40				√

Fig 4. Correlated, Associated, Associated-correlated and Independent Patterns

5 Experimental Results

All programs are written in Microsoft Visual C++ 6.0 running on Windows XP. And the Hardware configuration is as follows: Processor-Intel Core 2 Duo 2.4GHz, Main memory-4GB, and Hard disk space-500GB. Our experiments were performed on real data sets as shown in Table VI. Gazelle comes from click-stream data from http://gazelle.com and pumsb is obtained from http://www.almaden.ibm.com. On the other hand the Connect-4 dataset has been downloaded from website http://rchive.ics.uci.edu/.The gazelle is rather sparse in comparison with pumsb, which is very dense so that it produces many long frequent itemsets even for very high values of support. Table 5 shows the characteristics of these datasets.

Dataset	# Transactions	# Items	ATL/MTL**
Gazelle	59602	497	2.5/267
Pumsb	49046	2113	74/74
Connect-4	135,115	6500	8/35

Fig. 5. Characteristics of the datasets. ** Here, ATL is average & MTL is max transaction length.

We compared our results with existing algorithms [1, 4, 13]. We not only mined the frequent correlated patterns but also mine correlated, associated, associated-correlated and independent patterns synchronously. We named the algorithm presented at [1] as LAP2; LAP1 for [13] respectively; and CoMine for [4]. In the first experiment we observed the execution time of our IACAI algorithm on Connect-4 datasets (Fig. 6). In the second experiment we performed the execution time comparison between our IACAI algorithm, LAP1 [13] and CoMine algorithm [4] respectively. Fig 7(a) compared the execution time between CoMine and IACAI algorithm; on the other hand Fig 7(b) compared the execution time between our IACAI and LAP1. In the third experiment, we observed the execution time with change of min_corr_conf and min_sup (Fig 8(a)); and with change of min_all_conf and min_sup (Fig 8(b)) respectively on Connect-4 dataset. In both cases our IACAI algorithm outperforms LAP2.

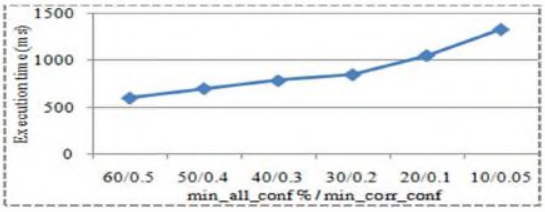

Fig 6. Runtime with change of min. corr-confidence and min. all-confidence on Connect-4 with min_sup=0.1%

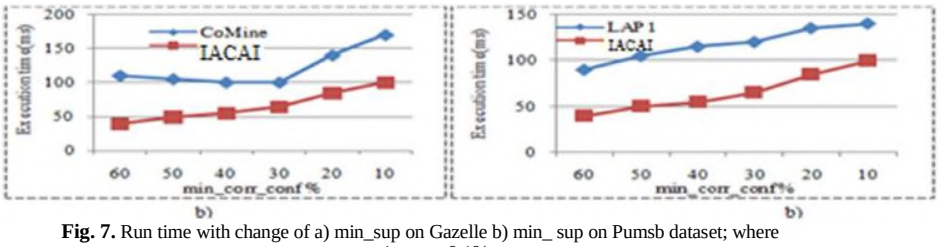

Fig. 7. Run time with change of a) min_sup on Gazelle b) min_ sup on Pumsb dataset; where min_sup=0.1%.

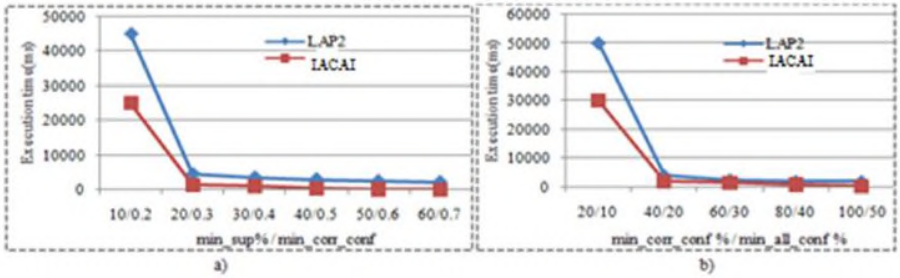

Fig. 8. a) Runtime with change of min_corr_conf & min_sup b) min_all_conf & min_sup on Connect-4.

6 Conclusion

In this paper we proposed an efficient "IACAI algorithm" that effectively mines the correlated, associated-correlated and independent patterns synchronously from an incremental transactional database. It also reduces the execution time as well as memory usages by removing '*null transactions*'. Experimental results show the correctness and scalability in terms of increasing load. We also showed how correlated pattern mining can be performed on top of an implementation of the FP-growth algorithm.

Acknowledgements

This work was supported by a grant from the NIPA (National IT Industry Promotion Agency, Korea) in 2012 (Global IT Talents Program).

References

1. Z. Zhou "Mining Frequent Independent Patterns and Frequent Correlated Patterns Synchronously" 5th International Conference on Fuzzy Systems and Knowledge Discovery, 2008.
2. B. Liu, W. Hsu, and Y. Ma, Pruning and Summarizing the Discovered Association. In Proc. 1999 ACM SIGKDD.
3. E. Omiecinski. Alternative interesting measures for mining associations. IEEE Trans. On KDE, 2003.
4. Y.K. Lee, W.Y. Kim, Y. D. Cai, J. Han. "CoMine: Efficient Mining of Correlated Patterns" (ICDM'03).
5. Khalil M. Ahmed, Nagwa M. El-Makky, and Yousry Taha. "Beyond Market Baskets: Generalizing Association Rules to Correlations". ACM SIGKDD Explorations, 2000.
6. H. T. Reynolds. The Analysis of Cross-Classifications, 1977.
7. Zhongmei Zhou, Chunshan Wang, Fengyi. Mining both associated and correlated patterns, ICCS 2006.
8. G. Piatetsky-Shapiro. Discovery, Analysis and Presentation ofStrong Rules. MIT Press, 1991.
9. J. Han and M. Kamber, Data Mining: Concepts and Techniques, 2nd Edition: Morgan Kaufmann, 2006.
10. S. Berchtold, D. A. Keim, and H. P. Kriegel, "The X-tree: An Index Structure for High-Dimensional Data," Readings in Multimedia Computing& Networking, 2001.
11. W. Y. Kim, Y.K. Lee, and Jiawei Han. "CCMine: Efficient Mining of Confidence-Closed Correlated Patterns". *PAKDD, 2004.*

12. Agrawal, R. and R. Srikant. Fast algorithms for Mining Association Rules. 20th VLD B Conf. 1994.
13. Z. Zhou1, C. Wang1 and Y. Feng: "Mining Both Associated and Correlated Patterns" ICCS, 2006.
14. Liu Yongmei, Guan Yong "Application in Market Basket Research Based on FP-Growth Algorithm", Proc. of the 2009 WRI World Congress on Computer Science and Information Engineering, USA.
15. Cong-Rui Ji, Zhi-Hong Deng "Mining Frequent Patterns without Candidate Generation", Fourth International Conference on Fuzzy Systems and Knowledge Discovery, 24-27 August 2007, China.
16. Yan Hu, Ruixue Han "An Improved Algorithm for Mining Maximal Frequent Patterns", International Joint Conference on Artificial Intelligence", 25-26 April, 2009, China.
17. Hui Chen "Mining Frequent Patterns in Recent Time Window over Data Streams", 10th IEEE International Conference on High Performance Computing and Communications, 25-27 September 2008, Dalian, China.
18. Leung C.K., S. Boyu Hao "Mining of Frequent Itemsets from Streams of Uncertain Data", Proceedings of the IEEE International Conference on Data Engineering, 29 March -2 April 2009, Shanghai, China.
19. Takeaki Uno, Tatsuya Asai, Yuzo Uchida, Hiroki Arimura "LCM: An Efficient Algorithm for Enumerating Frequent Closed Item Sets", Proceedings of Workshop on Frequent itemset Mining Implementations, Japan, Volume 54, Pages: 23.
20. A. Meenakshi, and Dr. K. Alagarsamy: "Efficient Storage Reduction of Frequency of Items in Vertical Data Layout" International Journal on Computer Science and Engineering (IJCSE), Vol. 3 No. 2 Feb 2011.