Global Journal of Computer Science and Technology
Volume 11 Issue 13 Version 1.0 August 2011
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

A Survey on Index Support for Item Set Mining

By T. Senthil Prakash, Dr. P. Thangaraj
PRIST University, Thanjavur

*Abstract-*It is very difficult to handle the huge amount of information stored in modern databases. To manage with these databases association rule mining is currently used, which is a costly process that involves a significant amount of time and memory. Therefore, it is necessary to develop an approach to overcome these difficulties. A suitable data structures and algorithms must be developed to effectively perform the item set mining. An index includes all necessary characteristics potentially needed during the mining task; the extraction can be executed with the help of the index, without accessing the database. A database index is a data structure that enhances the speed of information retrieval operations on a database table at very low cost and increased storage space. The use index permits user interaction, in which the user can specify different attributes for item set extraction. Therefore, the extraction can be completed with the use index and without accessing the original database. Index also supports for reusing concept to mine item sets with the use of any support threshold. This paper also focuses on the survey of index support for item set mining which are proposed by various authors.

Keywords : Frequent Item Set, Index Support, Vertical Index, MTR (Modified Transaction Reduction).

GJCST Classification : H.2.8

A Survey on Index Support for Item Set Mining

T. Senthil Prakash[α], Dr. P. Thangaraj[Ω]

Abstract - It is very difficult to handle the huge amount of information stored in modern databases. To manage with these databases association rule mining is currently used, which is a costly process that involves a significant amount of time and memory. Therefore, it is necessary to develop an approach to overcome these difficulties. A suitable data structures and algorithms must be developed to effectively perform the item set mining. An index includes all necessary characteristics potentially needed during the mining task; the extraction can be executed with the help of the index, without accessing the database. A database index is a data structure that enhances the speed of information retrieval operations on a database table at very low cost and increased storage space. The use index permits user interaction, in which the user can specify different attributes for item set extraction. Therefore, the extraction can be completed with the use index and without accessing the original database. Index also supports for reusing concept to mine item sets with the use of any support threshold. This paper also focuses on the survey of index support for item set mining which are proposed by various authors.

Keywords : Frequent Item Set, Index Support, Vertical Index, MTR (Modified Transaction Reduction).

I. Introduction

MANY real-life databases are updated through the blocks of occasionally received business information. In these databases, the content is occasionally updated through either insertion of new transaction blocks or removal of outdated transactions. Data can be portrayed as a sequence of incoming data blocks, where new blocks arrive periodically or old blocks are discarded. Examples of developing databases are transactional data from huge retail chains, web server logs, financial stock tickers, and call detail records [15]. Since the data develop overtime, algorithms must be developed to incrementally maintain itemset mining models.

Different categories of examination might be performed over such data like (a) mining all existing data (b) mining only the most current data (e.g., current week data), (c) mining periodical data (e.g., half-yearly data) and (d) mining preferred data blocks (e.g., data associated to the last month of this year and the 2nd month of last year). Consider for instance transactional data from huge retail chains, after shop closing on each day, a set of transactions is inserted into the database. In this situation, market analysts are interested in investigating different portions of the database to realize customer behaviors. For instance, they might be interested in investigating purchases before New Year or during summer vacation.

Physical analysis of these huge amount of information stored in modern databases is very difficult. A recognized data mining technique is association rule mining. It is able to discover all interesting relationships which are called as associations in a database. Association rules are very efficient in revealing all the interesting relationships in a relatively large database with huge amount of data. The large quantity of information collected through the set of association rules can be used not only for illustrating the relationships in the database, but also used for differentiating between different kinds of classes in a database. But the major difficulty in association rule mining is its complexity.

Association rule mining is a costly method that involves a significant amount of time and memory. Hence, suitable data structures and algorithms must be developed to efficiently carry out the task. Association rule mining is a two-step method: Frequent itemset mining [1] and generation of association rule. Since the first stage is the most computationally intensive knowledge mining task, research activity has been primarily focused on significant and effective algorithms to achieve this extraction task. The data to be analyzed is perhaps extracted from a database and stored into binary files. Numerous algorithms, both memory-based and disk-based are concentrated on specialized data structures and buffer management approach to efficiently mine the desired type of information from a flat dataset.

In this paper, itemset mining on developing databases has been focused. A database index [4] is a data structure that enhances the speed of information retrieval operations on a database table at very low cost and increased storage space. Indexes can be generated using one or more columns of a database table, offering the foundation effective access of ordered records. The disk space needed for storing the index is normally less than that needed by the table. Because indices typically include only the key-fields based on which the table is to be organized, and eliminate all the other details in the table. Thus providing the chance to store indices in memory for a table whose data is excessively large to store in memory.

Moreover, an index structure for extracting item set as sequence of data blocks. The index supports

Author [α] *: Ph.D Research Scholar, PRIST University, Thanjavur.* E-mail : jtyesp@yahoo.co.in
Author [Ω] *: Professor Head, Dept of CSE, Bannari Amman Institute of Technology, Sathyamangalam.*

user communication, where the user specifies many constraints for itemset extraction. It permits the mining of the complete set of itemsets which satisfy (a) time constraints and (b) support constraints. Since the index contains all feature potentially required during the mining task, the extraction can be carried out by means of the index, without accessing the database. The data representation is absolute, i.e., no support threshold is enforced throughout the index construction stage, to permit reusing the index for mining itemsets with any support threshold.

Constraints like support and confidence is not enforced throughout the index creation stage. Therefore, the extraction can be carried out using the index alone, without accessing the original database. As the databases are necessary in almost all the retail stores, super markets, etc., it is necessary to develop an approach for item set mining with the help of index support. There are many approaches available in the literature based on item set mining and index support.

II. Literature Survey

S. Sahaphong [1] suggested frequent itemsets mining using vertical index list. In this paper, the author proposes a new technique to mine all frequent itemsets [12] that executes database scanning only once to create data structure. This arrangement uses the conceptual of vertical data outline to include transaction data. The altering of minimal support is not effected by the data structure and rescan of database is not needed. This technique has the capability of discovering frequent itemsets without creation of candidate itemsets. It achieves absolute and accurate frequent itemsets. The experimental observation illustrates that this technique provides all definitions and accuracy of frequent itemsets.

Mining frequent itemsets from secondary memory was put forth by G. Grahne *et al.,* [2]. Mining frequent itemsets is main function of mining association rules, and it is understood algorithmically for main memory databases. In this approach, the authors examine techniques for mining frequent itemsets when the database or the data structures utilized in the mining are excessively large to fit in main memory. Experimental observations show that this technique reduces the required disk access by order of magnitude, and allow actual scalable data mining.

Yin-Ling Cheung *et al.,* [3] suggested mining frequent itemsets without support threshold: with and without item constraints. In traditional association rules mining, a minimum support threshold is considered to be available for mining frequent itemsets. But, fixing such a threshold is characteristically tough. This makes an additional practical difficulty; it is to mine N k-itemsets with the maximum supports for k equal to a certain kmax value. The final output is the N-most interesting itemsets. Normally, it is very simple for users to conclude N and kmax value. The author proposed two new approaches, namely LOOPBACK and BOMO. Experimental observation proves that this technique provides better result than the existing Itemset-Loop algorithm, and the output of BOMO can be an order of magnitude enhanced than the original FP-tree algorithm [24], still with the supposition of an optimally chosen support threshold. The author also proposed the mining of "N-most interesting k-itemsets with item constraints." This permits the user to denote different degrees of interestingness for dissimilar itemsets. Experimental observations show that this proposed Double FP-trees algorithm, is very effective in solving this problem which depends on BOMO.

E. Baralis *et al.,* [4] recommended itemset mining on indexed data blocks. Numerous attempts have been offered to combine data mining activities with relational DBMSs, but a correct incorporation into the relational DBMS kernel has been infrequently achieved. This paper suggested an innovative indexing method [16], which denotes the transactions in a succinct form, suitable for tightly incorporating frequent itemset mining in a relational DBMS. The data illustration is complete, i.e., no support threshold is imposed, with the intention to permit reusing the index for mining itemsets with any support threshold. In addition, a suitable structure of the stored information has been developed, in order to permit a selective access of the index blocks essential for the current extraction stage. The index has been executed into the PostgreSQL open source DBMS and utilizes its physical level access techniques. Many experiments have been done on several datasets, characterized by dissimilar data distributions. The implementation time of the frequent itemset [12] mining task exploiting the index is constantly similar with and sometime quicker than a C++ execution of the FP-growth technique accessing data stored on a flat file.

A fast algorithm for frequent itemset mining using FP-trees was proposed by G. Grahne *et al.,* [5]. Well-organized algorithms for mining frequent itemsets are vital for mining association rules, also for many additional data mining tasks. Techniques for extracting frequent itemsets have been employed by means of a prefix-tree structure called as FP-tree, implemented for storing compressed information regarding frequent itemsets. Many experimental observations have illustrated that these techniques execute very well. In this technique, the author proposed an innovative FP-array method that significantly decreases the necessity of traversing FP-trees, thus acquiring enhanced performance for FP-tree-based algorithms. The proposed approach works particularly well for sparse data sets. Additionally, the author proposed a novel technique for mining all, maximal, and closed frequent itemsets [21]. This approach uses the FP-tree data structure along with the FP-array method effectively and integrates several optimization techniques. Experimental result proves that this technique is the best for many cases. Although this approach takes much memory when the data sets are sparse but it is the fastest technique when the minimum support is low. Furthermore, this technique is the fastest techniques and uses less memory than previous techniques when the data sets are dense.

Xuegang Hu *et al.,* [6] suggested mining frequent itemsets using a pruned concept lattice. Extracting frequent itemsets is a critical step in association rule mining. On the other hand, most of the approaches which mine frequent itemsets examine databases numerous times, which reduces the efficiency. In this technique, the association among the concept lattice and frequent itemsets is used, and the method of pruned concept lattice (PCL) is established to characterize frequent itemsets in a specified database, and the scale of frequent itemsets is compressed efficiently. A technique for extracting frequent itemsets based on PCL is implemented, which prunes infrequent concepts appropriately and dynamically throughout the PCL's construction based on the Apriori property. The effectiveness of the approach is illustrated with experiments.

Improved paralleled algorithm for mining frequent item-set used in HRM was presented by XuePing Zhang *et al.,* [7]. This approach established the technique of multi-thread processing and a Multi-Threaded Paralleled frequent item-set extraction Algorithm - MTPA was implemented depending on FP-tree algorithm [24]. It has been implemented in an enterprise human resources management organization. Based on the experiments of paralleled mining by utilizing increasing multi-thread processing, it is confirmed that MTPA which on the circumstance of multi-core processors can enhance the efficiency of frequent item-set mining successfully.

Dong Liyan *et al.,* [8] proposed a novel method of mining frequent item sets. The goal of mining association rules is to determine the association relationship among the item sets from mass data. In a number of practical applications, its responsibility is mostly to support in decision-making. In this paper, the author proposed an association rule algorithm of mining frequent item sets, which establishes a new data structure and adopts compressed storage tree to develop the run performance of this algorithm. At last, the experiment indicates that the algorithm proposed in this paper has much more advantages in load balance and run time compared with most existing algorithms.

Guo Yi-ming *et al.,* [9] presented a vertical format algorithm for mining frequent item sets. Apriori is a traditional algorithm for association rules. With the purpose of obtaining the support degree of candidate sets, Apriori requires to scan the database for several times. This author proposed a new algorithm, which mine frequent item sets by means of vertical format. The proposed technique only needs to scan database one time. And in the follow-up data mining procedure, it can obtain new frequent item sets through `and operation' among item sets. This technique requires less storage space, and can enhance the efficiency of data mining.

Frequent closed informative itemset mining was suggested by Fu *et al.,* [10]. In modern years, cluster analysis and association investigation have attracted a lot of attention for large data analysis such as biomedical data analysis. In this paper, the author developed a novel algorithm of frequent closed itemset mining [21]. The algorithm deals with two challenges of data mining which are mining huge and elevated dimensional data and interpreting the outputs of data mining. Frequent itemset extraction is the main job of association analysis. The approach depends on concept lattice structure with the intention that frequent closed itemsets can be produced to decrease the complicity of extracting all common itemsets and each recurrent closed itemset has additional knowledge to make easy understanding of mining results. From this aspect, the paper also deals with the extension of the approach for cluster analysis. The experimental observation shows the efficiency of this algorithm.

A new approach of modified transaction reduction algorithm for mining frequent itemset was proposed by R.E. Thevar *et al.,* [13]. Association rule mining is to take out the interesting association and relation among the huge volumes of transactions. This procedure is segmented into two sub problem: first problem is to discover the frequent itemsets from the transaction and then the second problem is to build the rule from the mined frequent itemset. Frequent itemsets creation is the necessary and most time huge procedure for association rule mining. Currently, most well-organized apriori-like algorithms rely deeply on the minimum support constraints to prune the enormous amount of non-candidate itemsets. These algorithms store numerous unnecessary itemsets and transactions. In this paper, the authors proposed an innovative frequent itemsets creation algorithm called MTR-FMA (modified transaction reduction depends on frequent itemset mining algorithm) that sustains its performance even at relative low supports. The experimental output also proves that proposed MTR-FMA algorithm on an outset is quicker than high efficient AprioriTid and other algorithms.

Lei Wen *et al.,* [14] developed an efficient algorithm for mining frequent closed itemset. Association rule mining was a significant field of data mining investigation. Determining the potential frequent itemset was a vital step. The existed frequent itemset discovery algorithms could find out all the frequent itemset or maximal frequent itemset. N. Pasquier developed an innovative job of mining frequent closed itemset. The size of frequent closed itemset was much lesser than all the frequent itemsets and did not lose any information. In this paper, a new frequent closed itemset approach depends on the directed specified itemset graph. This approach can discover all the frequent closed itemset powerfully by using depth first search method. The experiment report confirms that it is effective for mining frequent closed itemsets.

A. Omari [17] developed a new temporal measure for interesting frequent itemset mining. Frequent itemset mining supports the data miner in searching for strongly associated items and transactions in huge transaction databases. Since the number of frequent itemsets is typically very large and uncontrollable for a human user, techniques for mining interesting rules have been developed to identify meaningful and summarized representation. Furthermore, many measures have been implemented in the literature to find out the interestingness of the rule. In this paper, the author establishes a novel temporal measure for interesting frequent itemset mining. This measure depends on the suggestion that interesting frequent itemsets are generally covered by many recent transactions. This measure diminishes the cost of searching for frequent itemsets by reducing the search interval. Furthermore, this measure can be used to improve the search strategy implemented by the Apriori algorithm.

A novel parallel frequent itemset mining algorithm was recommended by Chen Xiaoyun *et al.,* [18]. Frequent itemset mining is an essential and important matter in data mining field and can be utilized in many data mining tasks. The majority of these mining tasks need multiple passes over the database and if the database size is huge, which is typically the case; scalable elevated performance solutions involving multiple processors are needed. In this paper, the authors proposed a new equivalent frequent itemset mining approach which is called HPFP-Miner. The proposed technique depends on FP-Growth and establishes modest communication overheads by efficiently partitioning the list of frequent elements list over processors. The reports of experiment prove that HPFP-Miner has good scalability and performance.

E. Baralis *et al.,* [19] suggested itemset mining on indexed data blocks. This paper proposes a novel index, called I-Forest, to maintain data mining activities on developing databases, whose content is occasionally updated through insertion or deletion of data blocks. I-Forest permits the mining of itemsets from transactional databases such as transactional data from huge trade chains. Item, support and time constraints may be imposed throughout the extraction phase. The proposed index is a covering index that corresponds to transactional blocks in a succinct form and permits different kinds of examination (e.g., analyze quarterly data). Throughout the creation stage no support constraint is needed. Thus, the index offers an entire illustration of the developing data. The I-Forest index has been executed into the Post-greSQL open source DBMS and develop its physical level access methods. Experiments conducted on both sparse and dense data distributions. The implementation time of the frequent itemset mining task exploiting the index is always comparable with and for least support threshold quicker than the Prefix-Tree algorithm accessing static data on at file.

Jian Chen *et al.,* [20] provided a method for new items based on indexing techniques. The amount of information in the World Wide Web is growing very rapidly than our skill to process. Recommender systems provide knowledge innovation approaches to assist people find what they actually want. These approaches consist of collaborative filtering (CF), association rules detection and Bayesian networks, etc. But all of these methods have a significant disadvantage: items or pages which being uploaded to a site in recent times cannot be found. This is commonly called as the new item problem. The authors established a common structure for solving this difficulty and present a single index arrangement x-features-tree for using heuristic information retrieval approach to find the right items for the right users.

Mining association rules from XML data with index table was suggested by Xin-Ye Li *et al.,* [22]. Mining XML association rule is tackled with extra challenge because of the inherent flexibilities of XML in both arrangement and semantics. With the purpose of making mining XML association rule very efficient, this paper provides a new definition of transaction and item in XML environment, then construct transaction database depending on an index table. Based on the definition and the index table utilized for XML searching, it is easy to check the relation among the transaction and retrieve an item quickly. A high adaptive mining approach is also illustrated. By using this approach, mining rules can be processed with no assistance of interest associations specified by users and mining unknown rules. The effectiveness of these approaches is proved with the help of experiments on real-life data.

L. Golab *et al.,* [23] proposed indexing time method for evolving data with variable lifetimes. Numerous applications store data items for a pre-determined, fixed duration of time. Examples consist of sliding windows over online data streams, in which old data are thrown out as the window slides forward. Earlier researches on management of data with limited lifetimes have emphasized online query processed in main memory. In this approach, the authors concentrate on the difficulty of indexing time- developing data on disk for offline investigation [16]. With the intention of decreasing the I/O costs of index updates, existing work separates the data chronologically. Thus, only the previous separation is examined for expirations, only the youngest separations acquire insertions, and the remaining partitions in the middle are not processed. On the other hand, this result is based upon the hypothesis that the order in which the data are introduced is equivalent to the termination order, which means that the lifetime of each data item is the similar. In order to break this hypothesis, the authors reveal that the existing solutions no longer be relevant, and suggested a new index partitioning strategies that provide low update costs and quick access times.

E.J. Keogh *et al.,* [25] proposed an indexing scheme [11] for fast similarity search in large time series databases. This paper addresses the trouble of similarity searching in huge time-series databases. The authors proposed an innovative indexing approach that permits quicker retrieval. The index is produced by generating bins that include time series subsequences of roughly the similar shape. For every bin, this proposed approach can rapidly compute a lower bound on the distance among a given query and the most similar element of the bin. This bound permits to search the bins in greatest-first order, and to prune some bins from the search space without verifying the contents. Further speedup can be achieved by optimizing the data inside the bins in such a way that ignores the process of comparing the query to every item in the bin. This technique could be called as STB (Shape To Bit-vector) indexing, and experimentally confirm it on space telemetry, medical and synthetic data, demonstrating roughly an order-of-magnitude speedup.

III. Problems and Directions

Many problems are faced in the above discussed existing tecniques. To solve these problems many researches has to be done. The research should be focussed on the following areas.

a) Problems involved in Static Datasets

Most of the data mining algorithms available in the literature concentrate on the static datasets. These are the kind of the datasets which do not change over time. So, there is a need for some researches to focus on this static dataset problem and to develop some technique which supports the developing database based on index support to mine the item sets. Incremental update of the index will solve this difficulty. A technique should be developed to update the index whenever a new data is inserted.

b) Use of Correlation Information

It takes more time to read all the physical data blocks. Therefore, it is necessary to reduce the number of physical data blocks read throughout the mining process. For this purpose, correlated information can be stored in the same block. Also the system can be adopted for the tree based rule mining schemes and the system can be improved to detect positive and negative rule mining process.

IV. Conclusion

In recent business trends, it is necessary to transform the data available in a database into an informational advantage. Data mining is the process of extracting relatively useful information from a large data base. Since the usage of database increased in all the fields of research and also in the retail shop, there is a need for some techniques to make these available databases into a sequence of valuable information. For the quick retrieval of information indexes can be used in item set mining. It reduces the cost of storage and provides quick access of information. The available information in databases can be arranged based on some order, in such a way that should provide strong association among the data. Many algorithms and techniques have been discussed in the literature, which exists in the current approaches. The problems faced in the existing techniques and the directions will form the basis for the innovation of the new approaches. This survey will help the researchers to develop an efficient technique based on index support for item set mining.

References Références Referencias

1. S. Sahaphong, "Frequent itemsets mining using vertical index list," 2nd IEEE International Conference on Computer Science and Information Technology (ICCSIT), pp. 480 – 484, 2009.
2. G. Grahne and Jianfei Zhu, "Mining frequent itemsets from secondary memory," Fourth IEEE International Conference on Data Mining (ICDM '04), pp. 91 – 98, 2004.
3. Yin-Ling Cheung and Ada Wai-Chee Fu, "Mining Frequent Itemsets without Support Threshold: With and without Item Constraints," IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No. 9, pp. 1052-1069, 2004.
4. Baralis, T. Cerquitelli, and S. Chiusano, "Index Support for Frequent Itemset Mining in a Relational DBMS," Proceedings 21st International Conference on Data Engineering (ICDE), pp. 754 - 765, 2005.
5. Grahne and J. Zhu, "Fast algorithms for frequent itemset mining using FP-trees," IEEE Transactions on Knowledge and Data Engineering, Vol. 17, No. 10, pp. 1347 – 1362, 2005.
6. Xuegang Hu, Wei Liu, Dexing Wang and Xindong Wu, "Mining Frequent Itemsets Using a Pruned Concept Lattice," Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007), Vol. 3, pp. 606 – 610, 2007.
7. XuePing Zhang, YanXia Zhu and Nan Hua, "Improved paralleled algorithm for mining frequent item-set used in HRM," Seventh International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), Vol. 6, pp. 2830 – 2833, 2010.
8. Dong Liyan, Liu Zhaojun, Shi Mo, Yan Pengfei, Tian Zhuo and Li Zhen, "A novel method of mining

frequent item sets," IEEE International Conference on Information and Automation (ICIA), pp. 173-178 , 2010.

9. Guo Yi-ming and Wang Zhi-jun, "A vertical format algorithm for mining frequent item sets," 2nd International Conference on Advanced Computer Control (ICACC), Vol. 4, pp. 11 – 13, 2010.
10. Fu, Huaiguo Foghlu, Micheal O. Donnelly and Willie, "Frequent Closed Informative Itemset Mining," International Conference on Computational Intelligence and Security, pp. 232 – 236, 2007.
11. Gudes, "A uniform indexing scheme for object-oriented databases," Proceedings of the Twelfth International Conference on Data Engineering, pp. 238 – 246, 1996.
12. Grahne and J. Zhu "Efficiently using prefix-trees in mining frequent itemsets" In FIMI, 2003.
13. R.E. Thevar and R. Krishnamoorthy, "A new approach of modified transaction reduction algorithm for mining frequent itemset," 11th International Conference on Computer and Information Technology (ICCIT 2008), pp. 1 – 6, 2008.
14. Lei Wen, "An efficient algorithm for mining frequent closed itemset," Fifth World Congress on Intelligent Control and Automation (WCICA 2004), Vol. 5, pp. 4296 – 4299, 2004.
15. M. El-Hajj and O. R. Zaiane, "Inverted matrix: Efficient discovery of frequent items in large datasets in the context of interactive mining" In Association for Computing Machinery's Special Interest Group on Knowledge Discovery and Data Mining, 2003.
16. Ramesh, W. Maniatty and M. Zaki, "Indexing and data access methods for database mining," In Data Mining and Knowledge Discovery, 2002.
17. Omari, "A new temporal measure for interesting frequent itemset mining," 2nd IEEE International Conference on Information Management and Engineering (ICIME), pp. 425 – 429, 2010.
18. Chen Xiaoyun, He Yanshan, Chen Pengfei, Miao Shengfa, Song Weiguo and Yue Min, "HPFP-Miner: A Novel Parallel Frequent Itemset Mining Algorithm," Fifth International Conference on Natural Computation (ICNC '09), Vol. 3, pp. 139 – 143, 2009.
19. Baralis, T. Cerquitelli and S. Chiusano, "Itemset Mining on Indexed Data Blocks," 3rd International IEEE Conference on Intelligent Systems, pp. 820 – 825, 2006.
20. Jian Chen, Jian Yin and Jin Huang, "Recommendation of new items based on indexing techniques," Proceedings of 2004 International Conference on Machine Learning and Cybernetics, Vol. 2, pp. 1168 – 1172, 2004.
21. Jianyong Wang, J. Han, Y. Lu and P. Tzvetkov, "TFP: an efficient algorithm for mining top-k frequent closed itemsets" IEEE Transactions on Knowledge and Data Engineering, Vol. 17, No. 5, pp. 652 – 663, 2005.
22. Xin-Ye Li, Jin-Sha Yuan and Ying-Hui Kong, "Mining Association Rules from XML Data with Index Table," International Conference on Machine Learning and Cybernetics, Vol. 7, pp. 3905 – 3910, 2007.
23. L. Golab, P. Prahladka and M.T. Ozsu, "Indexing Time-Evolving Data With Variable Lifetimes," 18th International Conference on Scientific and Statistical Database Management, pp. 265 – 274, 2006.
24. Liu, H. Lu, Y. Xu and J. X. Yu, "Ascending frequency ordered prefix-tree: Efficient mining of frequent patterns", In Database Systems for Advanced Applications (DASFAA), 2003.
25. E.J. Keogh and M.J. Pazzani, "An indexing scheme for fast similarity search in large time series databases," Eleventh International Conference on Scientific and Statistical Database Management, pp. 56 – 67, 1999.