A New Dimension to Improve the Query Performance using Disjoint Set Theory

M. Ranjit Reddy
Assistant professor
SRIT, Anantapur

M. Narasimhulu
Assistant professor
SRIT, Anantapur

M. Ashok
Senior Lecturer
RIT, Chennai

ABSTRACT

An index improves the speed of data retrieval operations on a table. Index Management includes Creation, insertion, deletion & Updation. Reducing the access time of index even for more number of transactions will be our objective. Algorithms efficiency fell down when the index updation is occurring in the existing literature. Managing index is a tough task when the size of database increases. We are organizing clusters using disjoint set theory and a ranking algorithm. It will improve the performance of querying.

Keywords: index, set, rank, query.

1. INTRODUCTION

Association rule mining discovers correlations among data items in a transactional database D. Each transaction in D is a set of data items. Association rules are usually represented in the form A→ B, where A and B are item sets, i.e., sets of data items. Item sets are characterized by their frequency of occurrence in D, which is called support. Research activity usually focuses on defining efficient algorithms for item set extraction, which represents the most computationally intensive knowledge extraction task in association rule mining [2]. The data to be analyzed is usually stored into binary files, possibly extracted from a DBMS. Most algorithms [2], [3], [4], [5], [6], [7], [8] exploit ad hoc main memory data structures to efficiently extract item sets from a flat file. Recently, disk-based extraction algorithms have been proposed to support the extraction from large data sets [7], [9], [10], but still dealing with data stored in flat files. To reduce the computational cost of item set extraction, different constraints may be enforced [11], [12], [13], [14], among which the most simple is the support constraint, which enforces a threshold on the minimum support of the extracted item sets. Relational DBMSs exploit indices, which are ad hoc data structures, to enhance query performance and support the execution of complex queries. In this paper, we propose a similar approach to support data mining queries. The IMine index (Item set-Mine index) is a novel data structure that provides a compact and complete representation of transactional data supporting efficient item set extraction from a relational DBMS.

2. DATA ACCESS METHODS

The existing three data Access methods [1] to load from the IMine index the following projections of the original database:

1) Frequent-item based projection, to support projection-based algorithms (e.g., FP-growth [3]).

2) Support-based projection, to support level based (e.g., APRIORI [1]), and array-based (e.g., LCM v.2 [15]) algorithms.

3) Item-based projection, to load all transactions where a specific item occurs,

enabling constraint enforcement during the extraction process.

The existing methods support minimum support of index updation. Index updation can't be incremental in the survey.

3. PROPOSED ALGORITHM

In this section we discussed the algorithm that rectifies the problem which is encountered in the existing system.

Step1: Find out the mostly accessed data with reference to the table.

Step2: Frame separate set which are accessed frequently.

Step3: Check for intersection of subsets such that commonly accessed elements should not be present.

Step4: determine the Rank based on cardinality of the sets.

Step5: Repeat Step1 to 3 till all the data items are iterated.

4. EXPERIMENTAL RESULTS

Table 1.1 illustrates the representation of item sets. Table 1.2 depicts the frequently accessed items from Table 1.1. Table 1.3 describes our set Ranking algorithm on the available item sets. Results of Table 1.1, 1.2 and 1.3 are summarizing the Fig.1 which implies the updation ratio clearly.

Table 1.1 Sample Data

Transactions ID	ITEMS ID					
001	A	B	D	F	K	P
002	B	C	H	E	G	I
003	I	J	A	C	D	F
004	D	E	F	G	I	
005	F	A	B	D	H	I
006	J	F	C	D	A	

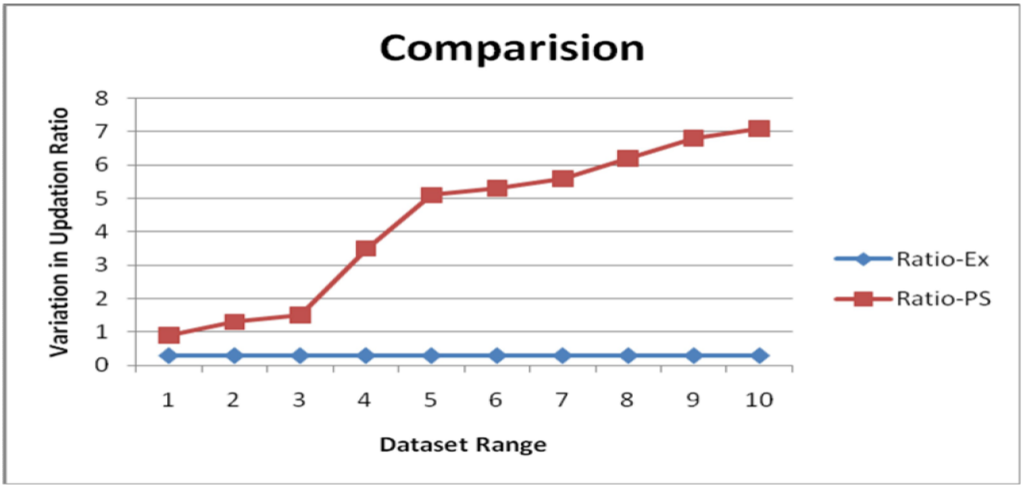
Table 1.2 Frequent sample

ITEMS ID	No. Of Times Accessed
{A, I}	4
{B, C}	3
{D, F}	5
{E, G, H, J}	2
{K, P}	1

Table 1.3 Rank Calculations

ITEMS ID	No. Of Times Accessed	Rank
{D, F}	5	1
{A, I}	4	2
{B, C}	3	3
{E, G, H, J}	2	4
{K, P}	1	5

Fig.1- Comparison



5. CONCLUSIONS

We have designed a Set Ranking algorithm based on Disjoint Sets. We revealed better index updation rate. On extending our work, we planned to fix the index updation time as the tradeoff.

6. ACKNOWLEDGMENTS

We wish to thank Mr. A Suresh Babu, Assistant Professor, JNTU Pulivendula, Kadapa, for his insightful comments and suggestions.

7. REFERENCES

[1] Elena Baralis, Tania Cerquitelli, and Silvia Chiusano, "IMine: Index Support for Item Set Mining," Proc. IEEE Trans. Knowledge and Data Eng., vol. 21, no. 4, April 2009.

[2] R. Agrawal and R. Srikant, "Fast Algorithm for Mining Association Rules," Proc. 20th Int'l Conf. Very Large Data Bases (VLDB '94), Sept. 1994.

[3] R. Agrawal, T. Imilienski, and A. Swami, "Mining Association Rules between Sets of Items in Large Databases," Proc. ACM SIGMOD '93 May 1993.

[4] J. Han, J. Pei, and Y. Yin, "Mining Frequent Patterns without Candidate Generation," Proc. ACM SIGMOD, 2000.

[5] H. Mannila, H. Toivonen, and A.I. verkamo, "Efficient Algorithms for discovering Association Rules," Proc. AAAI Workshop Knowledge Discovery in Databases (KDD '94), pp. 181-192, 1994.

[6] A. Savasere, E. Omiecinski, and S.B.Navathe, "An Efficient Algorithm for Mining Association Rules in Large Databases," Proc. 21st Int'l Conf. Very Large Data Bases (VLDB '95), pp. 432-444, 1995.

[7] H. Toivonen, "Sampling Large Databases for Association Rules," Proc. 22nd Int'l Conf. Very Large Data Bases (VLDB '96), pp. 134-145, 1996.

[8] M. El-Hajj and O.R. Zaiane, "Inverted Matrix: Efficient Discovery of Frequent Items in Large Datasets in the Context of Interactive Mining," Proc. Ninth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (SIGKDD),2003.

[9] G. Grahne and J. Zhu, "Mining Frequent Itemsets from Secondary Memory," Proc. IEEE Int'l Conf. Data Mining (ICDM '04), pp. 91-98, 2004.

[10] G. Ramesh, W. Maniatty, and M. Zaki, "Indexing and Data Access Methods for Database Mining," Proc. ACM SIGMOD Workshop Data Mining and Knowledge Discovery (DMKD), 2002.

[11] Y.L. Cheung, "Mining Frequent Itemsets Without Support Threshold: With and without Item Constraints," IEEE Trans. Knowledge and Data Eng., vol. 16, no. 9, pp. 1052-1069, Sept. 2004.

[12] G. Cong and B. Liu, "Speed-Up Iterative Frequent Itemset Mining with Constraint Changes," Proc. IEEE Int'l Conf. Data Mining (ICDM '02), pp. 107-114, 2002.

[13] C.K.-S. Leung, L.V.S. Lakshmanan, and R.T. Ng, "Exploiting Succinct Constraints Using FP-Trees," SIGKDD Explorations Newsletter, vol. 4, no. 1, pp. 40-49, 2002.

[14] R. Srikant, Q. Vu, and R. Agrawal, "Mining Association Rules with Item Constraints," Proc. Third Int'l Conf. Knowledge Discovery and Data Mining (KDD '97), pp. 67-73, 1997.

[15] T. Uno, M. Kiyomi, and H. Arimura, "LCM ver.2: Efficient Mining Algorithms for Frequent/Closed/Maximal Itemsets," Proc. IEEE ICDM Workshop Frequent Itemset Mining Implementations (FIMI), 2004.