International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:4, 2007

A Patricia-Tree Approach for Frequent Closed Itemsets

Moez BEN HADJ HAMIDA and Yahya SLIMANI

***Abstract*—**In this paper, we propose an adaptation of the Patricia-Tree for sparse datasets to generate non redundant rule associations. Using this adaptation, we can generate frequent closed itemsets that are more compact than frequent itemsets used in Apriori approach. This adaptation has been experimented on a set of datasets benchmarks.

***Keywords*—**Datamining, Frequent itemsets, Frequent closed itemsets, Sparse datasets.

I. Introduction

EFFICIENT algorithms for mining frequent itemsets are crucial for mining association rules. Methods for mining frequent itemsets and for iceberg data cube computation have been implemented using a prefix-tree structure, namely FP-Tree, for storing compressed informations about frequent itemsets. As pointed out by Han [6] the FP-Tree loses his compactness on sparse datasets, but they still use this structure for mining frequent closed itemsets. In this paper we propose an enhanced version of Patricia-Tree structure that reduces considerably the size taken by an FP-Tree and the build time. This structure is especially suitable for sparse datasets.

Datamining algorithms based on frequent itemsets like Apriori suffer from two drawbacks: (i) multiple scans to a dataset to compute the frequency of itemsets; (ii) high number of generated association rules. To avoid these two drawbacks, many solutions are used, like formal analysis concepts, parallelism, data structures adapted to datamining and so on. In this paper, we propose an adaptation of the Patricia-Tree structure for sparse datasets to find frequent closed itemsets. Then, we experiment this new structure on different datasets and compare it with the FP-Tree structure. The remaining of the paper is as follows: in section 2, we present briefly the main approaches used to generate association rules. Section 3 presents and compares the FP-Tree structure and the proposed adaptive Patricia-Tree for frequent closed itemsets. In section 4, we discuss some experimental results on dense and sparse datasets. Section 5 concludes the paper and gives some extensions of our work.

II. Previous Work

A. Apriori-based Algorithms

The most generic frequent patterns mining algorithm is Apriori [2]. This algorithm is based on frequent itemsets that are generated from candidate itemsets [2]. Using this approach, a number of Apriori-based algorithms [1,3,4] have been developed. Among these algorithms, only those use a Hash-tree representation of the database are efficient [1].

B. Pattern Growth Algorithms

Han proposes a new technique for mining frequent itemsets without generating candidate itemsets [5]. It defines two new data structures: frequent pattern tree or FP-Tree to compact dense datasets and H-struct [6] to deal with sparse datasets solely. Later, Pietracaprina and Zandolin have proposed to use a better compressed tree, called Patricia-Tree [7].

C. Closed Itemsets Mining

To reduce the huge number of rules produced by algorithms based on frequent itemsets, Pasquier [8] proposes to generate only frequent closed (i.e. non redundant) patterns. The algorithms that generate frequent closed itemsets use either item-based data structures [8,9] or the FP-Tree structure [10,11].

III. Patricia-Tree vs FP-Tree

In this section, we compare the Patricia-Tree structure with PF-Tree in order to determine what is more accurate for different databases (dense or sparse).

A. FP-Tree

The FP-Tree structure consists of a set of prefix subtrees under a root node labeled as "*null*" and a header table containing frequent items [5]. Every header table entry points a node in the FP-Tree carrying the same item name and every node on the FP-Tree points to the next occurrence of this item.

B. Patricia-Tree

A Patricia-Tree is a compressed FP-Tree. We keep the same representation as an FP-Tree but we merge every parent node with his single child node when they have the same support value [7]. Contrarily to an FP-Tree node that represents a single item a Patricia-Tree node can represent several items.

C. Comparison

As pointed out by Han [6], the FP-Tree loses his compactness on sparse datasets, but they still use this structure

Manuscript received January 19, 2005.

M. BHH is with the Department of Computer Science of Faculty of Sciences of Tunis, Tunisia (e-mail: moez.belhadj@ gawab.com).

Y. S. is with the Department of Computer Science of Faculty of Sciences of Tunis, Tunisia (correspondence author, phone:+21698537921, Fax:+21670860437, e-mail: yahya.slimani@fst.rnu.tn).

for mining frequent closed itemsets. The compactness of the FP-Tree is materialized by the merge of common prefixes for dense datasets. But for the sparse ones there is few prefixes shared by the transactions. In this case the number of FP-Tree nodes becomes close to the original dataset size (i.e. the sum of all transaction lengths).

For these reasons, we propose to adapt the Patricia-Tree proposed in [7] to generate frequent closed itemsets [13]. For example, consider the dataset represented by table 1 with *minsup* set to 3. Its representations by the FP-Tree and the Patricia-Tree are given respectively by figures 1 and 2.

TABLE 1
SAMPLE DATASET D

TID	Items
1	A B D E F G H I
2	B C E L
3	A B D F H L
4	A B C D F G L
5	B G H L
6	A B D F I

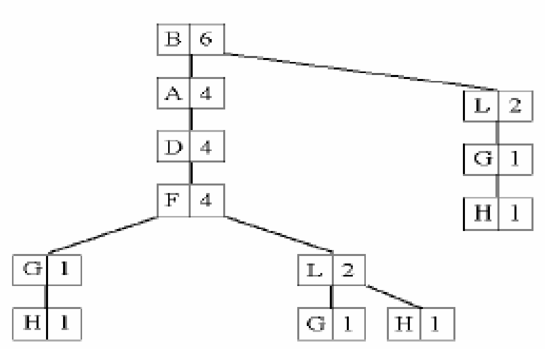

FIG 1. FP-TREE FOR DATASET D

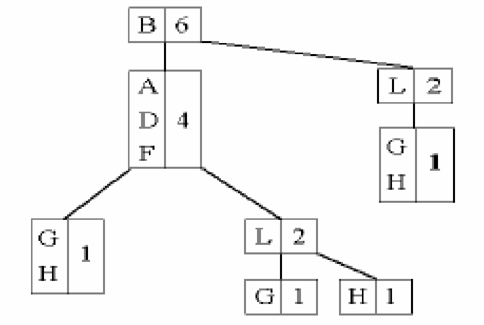

FIG 2. PATRICIA-TREE FOR DATASET D

The above figures show that a Patricia-Tree is more compact than an FP-Tree. In fact, a dataset consisting of M transactions with aggregate size N can be represented through a Patricia-Tree of size at most equal to N + *O*(M) [7]. But for an FP-Tree when the dataset is highly sparse, the number of nodes may be close to N * M.

IV. EXPERIMENTAL RESULTS

In this section, we discuss results of several experimentations of our proposed data structure for different datasets. The goal of these experimentations is to find out the extent of different dataset properties that could affect the performance of Patricia-Tree and its relative performance with the FP-Tree [13].

Experiments were performed on a 500MHz Pentium PC with 320MB of memory, running on RedHat Linux 8.2.

Our version of Patricia-Tree was implemented in C and the FP-Tree was coded in C++ by Zhu [12].

For our experimentations, we have used several real and synthetic database benchmarks, publicly available at the FIMI[1] workshop site. The PUMSB dataset contains census data. The MUSHROOM database contains characteristics of various species of mushrooms. The CONNECT dataset is derived from its game steps. The synthetic datasets T40I10D100K and T20I10D10K, obtained from IBM Almaden generator, mimic the transactions in a retailing environment.

Table 2 gives the characteristics of the real and synthetic datasets used in our evaluation. It shows the number of items, the average transaction length, the standard deviation of transaction lengths, and the number of records in each database.

The first experiment compares the FP-Tree and the Patricia-Tree build times for the different datasets.

As shown in Figures 3 and 5, these structures perform the same build time for real datasets. But Figure 4 shows that the build time of a Patricia-Tree is about two orders of magnitude faster than FP-Tree for synthetic datasets.

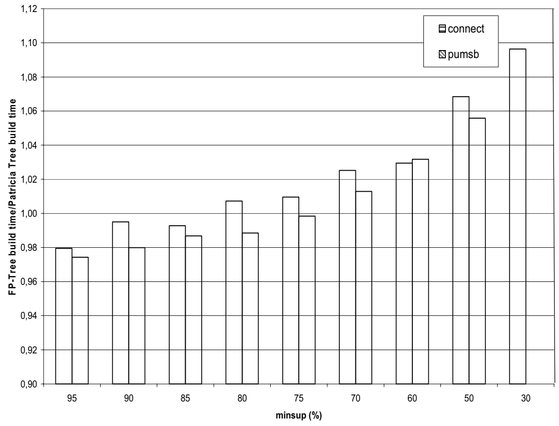

FIG 3. FP-TREE BUILD TIME / PATRICIA-TREE BUILD TIME FOR REAL DATASETS.

[1] http ://fimi.cs.helsinki.

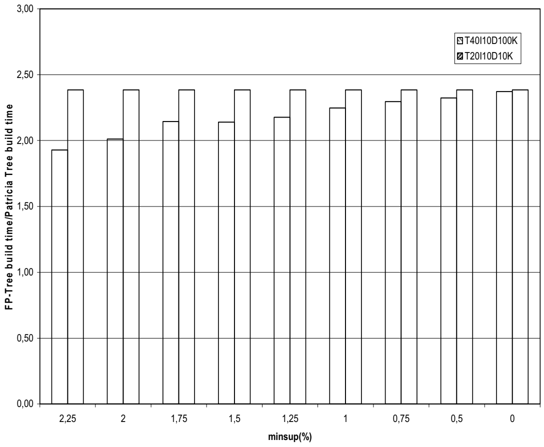

FIG 4. FP-TREE BUILD TIME/ PATRICIA-TREE BUILD TIME FOR SYNTHETIC DATASETS.

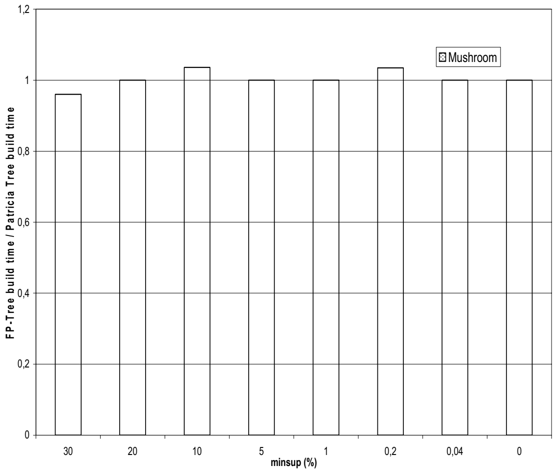

FIG 5. FP-TREE BUILD TIME/ PATRICIA-TREE BUILD TIME FOR DATASET MUSHROOM.

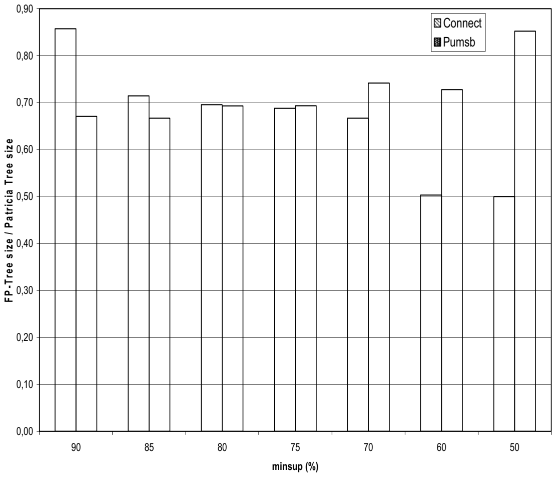

FIG 6. FP-TREE SIZE/PATRICIA-TREE SIZE FOR REAL DATASETS.

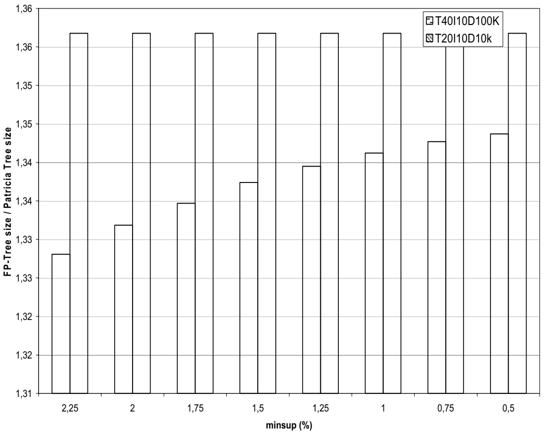

FIG 7. FP-TREE SIZE /PATRICIA-TREE SIZE FOR SYNTHETIC DATASETS.

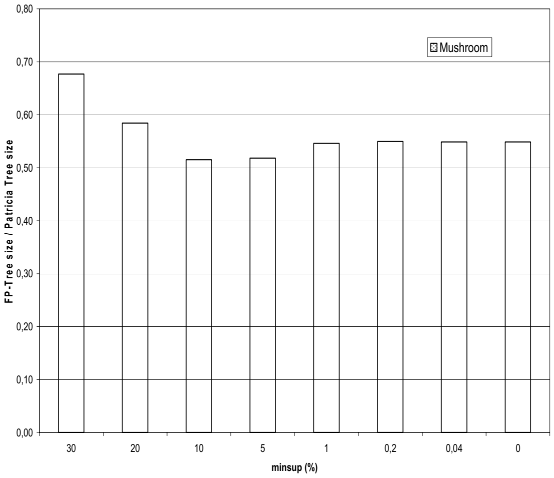

Fig 8. FP-TREE SIZE /PATRICIA-TREE SIZE FOR DATASET MUSHROOM.

The second experiment measures the memory space needed by FP-Tree and Patricia-Tree for the benchmark datasets. On real datasets Patricia-Tree consume more space than FP-Tree due to the over head taken by the number of items on a node and other data needed to tree traversal (Figures 6 and 8). For sparse datasets, Figure 7 shows that Patricia-Tree is more memory efficient.

V. CONCLUSION

We proposed an adaptation of the Patricia-Tree structure to find frequent closed itemsets. This adaptation allows to define more efficient datamining algorithms than those used FP-Tree structure. The experimentations of this adapted Patricia-Tree structure have showed that it is more suitable for sparse datasets.

In the future, we plan use this same structure to find sequential patterns and closed sequential patterns.

References

[1] J.S. Park, M.S. Chen and P.S. Yu, "An Effective Hash Based Algorithm for Mining Association Rules," in *Proc. 5th SIGMOD Intl. Workshop. Management of Data*, California, 1995, pp. 175–186.

[2] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules*. 20th Intl. Conf. Very Large Data Bases*, Santiago, 1994, pp. 487–499.

[3] A. Savasere, E. Omiecinski and S. Navathe, "An Efficient Algorithm for Mining Association Rules in Large Databases," in *Proc. 21th Intl. Conf. Very Large Data Bases*, Santiago, 1995, pp. 487–499.

[4] S. Brin, R. Motwani, J. Ullman and S. Tsur, "Dynamic itemset counting and implication rules for market basket data," in *Proc.* 7th *SIGMOD Intl. Workshop. Management of Data*, Arizona, 1997, pp. 255–264.

[5] K. Wang, L. Tang, J. Han and J. Liu, "Top Down FP-Growth for Association Rule Mining," in *Proc. 6 th Pacific-Asia Conf. Advances in Knowledge Discovery and Data Miningy*, Taipei, 2002, pp. 334–370.

[6] J. Pei, J. Han, H. Lu, S. Nishio, S. Tang and D. Yang, "H-Mine: Hyper-Structure Mining of Frequent Patterns in Large Databases," in *Proc. 1st IEEE Intl. Conf. Data Mining*, California, 2001, pp. 441–448.

[7] A. Pietracaprina and D. Zandolin, "Mining Frequent Itemsets using Patricia Tries," in *Proc.* 1st *FIMI Workshop. Frequent Itemset Mining Implementations*, Florida, 2003.

[8] N. Pasquier, Y. Bastide, R. Taouil and L. Lakhal, "Discovering Frequent Closed Itemsets for Association Rules," in *Proc.* 7th *Intl. Conf. Database Theory*, Jerusalem, 1999, pp. 398–416.

[9] M.J. Zaki and C. Hsiao, "CHARM: An Efficient Algorithm for Closed Itemset Mining," in *Proc.* 2nd *SIAM Intl. Con. Data Mining*, Virginia, 2002, pp. 398–416.

[10] J. Pei, J. Han, R. Mao, "CLOSET: An Efficient Algorithm for Mining Frequent Closed Itemsets," in *Proc. 9th SIGMOD Intl. Workshop. Data Mining and Knowledge Discovery,* Dallas, 2000, pp. 11–20.

[11] J. Wang, J. Han and J. Pei, "CLOSET+: Searching for the Best Strategies for Mining Frequent Closed Itemsets," in *Proc.* 12th *Intl. Conf. Knowledge Discovery and Data Mining*, Washington, 2003, pp. 236–245.

[12] G. Grahne and J. Zhu, "Efficiently using prefix-trees in mining frequent itemsets," in *Proc.* 1st *FIMI Workshop. Frequent Itemset Mining Implementations*, Florida, 2003.

[13] M. Ben Hadj Hamida, "Patricia-Tree based algorithm to find frequent closed itemsets," Master. dissertation, Dept. Comp. Sci., Faculty of Sciences of Tunis., Tunis, Tunisia, 2005.

Moez Ben Hadj Hamida was born in Tunis on May 15, 1979. He studied at the Department of Computer Science at the Faculty of Sciences of Tunis from 1999 to 2005. He received the B.Sc.(Eng.) and Master degrees from the Faculty of Sciences of Tunis, in 2003 and 2005, respectively. His currently prepares its Ph.D thesis on Computer Science at the Faculty of Sciences of Tunis.

He is currently lecturer assistant at the Department of Computer Science of Faculty of Sciences of Tunis, Tunisia. These research activities concern datamining, parallelism and grid computing.

Yahya Slimani was born in Oujda on March 19, 1951. He studied at the Computer Science Institute of Alger's (Algeria) from 1968 to 1973. He received the B.Sc.(Eng.), Dr Eng and Ph.D degrees from the Computer Science Institute of Alger's (Algeria), University of Lille (French) and University of Oran (Algeria), in 1973, 1986 and 1993, respectively. His currently Professor at the Department of Computer Science of Faculty of Sciences of Tunis. These research activities concern datamining, parallelism, distributed systems and grid computing.

Dr. Slimani has published more than 80 papers from 1986 to 2005. He contributed to Parallel and Distributed Computing Handbook, Mc Graw-Hill, 1996.

He is currently President of African Conference on Computer Science. He joined the Editorial Boards of the Information International Journal in 2000.