Improving Mining Efficiency: A New Scheme for Extracting Association Rules

Aiman Moyaid Said[a], P D D. Dominic[b], Azween B Abdullah[c]

Department of Computer and Information Science
Universiti Teknologi PETRONAS
[a]Aiman.moyaid@yahoo.com [b]dhanapal_d@ petronas.com.my [c]azweenabdullah@petronas.com.my

ABSTRACT

In the age of information technology, the amount of accumulated data is tremendous. Extracting the association rule from this data is one of the important tasks in data mining. Most of the existing association rules in algorithms typically assume that the data set can fit in the memory. In this paper, we propose a practical and effective scheme to mine association rules from frequent patterns, called Prefixfoldtree scheme (PFT scheme). The original dataset is divided into folds, and then from each fold the frequent patterns are mined by using the tree projection approach. These frequent patterns are combined into one set and finally interestingness constraints are used to extract the association rules. The experiments will be conducted to illustrate the efficiency of our scheme.

Keywords
Data mining, Association Rules, Prefixfoldtree scheme, Frequent patterns, Fold.

1.0INTRODUCTION

The data mining has been considered a promising ground in the crossroads of databases, machine learning, and artificial intelligence (Mannila et al., 1994; Srikant et al., 1995). In Data mining, association rule mining has been one of the most fashionable subjects which can be defined as discovering meaningful patterns from large collections of data. When the association rule mining was first presented in (Agrawal et al., 1993) , it was developed to discover an important association rule between items in basket data in order to afford valuable information to the management of the retail store (such as how to design coupons, what to put on sale, how to place goods on shelves) to maximize profit. Association rules are employed in many application areas which play a significant role in various industries, such as the education industry (Encheva et al., 2006) , banking industry (Aggelis, 2004) , medical industry (Carlos et al., 2006) , and many others.

To tackle the problem of memory lack and the quality of the association rules, this paper presents a new scheme for mining association rules from frequent patterns, called Prefixfoldtree scheme (PFT scheme). The original dataset is divided into folds, and then from each fold the frequent patterns are mined using the tree projection approach. These frequent patterns are combined into one set and finally interestingness constraints are used to extract the association rules.

The paper is organized as follows. Section 2 briefly discusses the related works; Section 3 gives the main principles of the scheme. Finally, this paper is concluded in section 4.

2.0 RELATED WORKS

The route of extract the association rules can be viewed as two-phases: the first phase is to mine all frequent patterns; each of these patterns will happen at least as frequently as preset minimum support count (min_sup). The second phase is to produce strong association rules from the frequent patterns; these rules must assure minimum support and minimum confidence. The performance of discovering association rules is largely determined by the first phase, (Han et al., 2006).

FP-growth approach for mining frequent itemsets without candidate generation was proposed by (Han et al., 2000). Its scalable frequent patterns mining method has been proposed as an alternative to the Apriori-based approach in (Agrawal et al., 1994). This algorithm creates a compact tree-structure, FP-tree, representing frequent patterns, which moderates the multi-scan problem and improves the candidate itemset generation. This algorithm is faster than others in the literature, this reported by the authors of this algorithm. Many variations of FP-tree and FP-growth were introduced. (Liu et al., 2003; Grahne et al., 2005; Gao, 2007), adapted the similar approach of (Han et al., 2000) for mining the

frequent itemsets from the transactional database. The authors reported that these algorithms are more efficient than FP-growth.

In addition to mining the frequent patterns a lot of work was proposed in the literature to find the solution for the memory-based problems due to the huge size of the database. (Savasere et al., 1995; Wang et al., 2006; Nguyen et al., 2008), they presented different approaches depending on partition-based. These approaches outperform the existing algorithms as reported by their authors.

3.0 THE PRINCIPLES OF THE SCHEME

3.1 Related Definitions:

To enrich our understanding of the main theme of this paper we will introduce the following definitions:

Definition 1: The transaction database retains transaction records. Every record has a unique identifier TID and all the items included in a transaction record are listed in ascending order. Table 1 displays a simple transaction database named D.

The identifier T100 is the first transaction record in D. Transaction T100 contains the items I1, I2, I5, which are listed in order of item suffixes (e.g. 1, 2,5).

Definition 2: The item is frequent if its support count (which refers to the number of transactions that include a particular itemset) is greater than or equal to the predetermined support threshold (min_sup). Otherwise it is infrequent.

Table 1:*Transaction database*

TID	List of Item_IDs
T100	I1,I2,I5
T200	I2,I4
T300	I2,I3
T400	I1,I2,I4
T500	I1,I3
T600	I2,I3
T700	I1,I3
T800	I1,I2,I3,I5
T900	I1,I2,I3

Definition 3: The transaction database D with particular number of records RN is logically divided into number of non-overlapping folds. If RN is odd then the number of the transaction (RN-1/NF), where NF (the number of folds), the memory size determines the number of folds, and the remaining transactions are added to the last Fold. In other words if RN is even then the number of the fold is (RN/NF).

Definition 4: PFT is the prefix tree constructed for every fold in the database (fold$\subseteq$D). Mainly it consists of a number of tree nodes and the root of the tree is labeled as NULL. Every node has N sub-nodes (n=1,2,3,…). The node is called leaf node when it has no child. The nodes are labeled with the name of the items in the transaction database.

3.2 PFT Scheme Approach:

This approach adapts FP-tree structure from (Gao, 2007) for constructing the Prefixfoldtree structure for each fold (fold$\subseteq$D). Any combination of nodes (the root is not included) on a branch from the leaf to the root is known as combination C. An example of a combination will be like $\{I5,I1\}_{[2]}$, I5,I1 present the labels of the nodes combined. A number in the bracket on the right is the frequency of this combination. Figure 1 illustrates the combination method. I4, I5 are leaf nodes in Figure 1 because the frequency of I5=1 and as there are three nodes I1,I2,I5 on the branch, four compositions are constructed. These combinations are $\{I5,I2\}_{[1]}$, $\{ I5 , I1 \}_{[1]}$,$\{I5,I2,I1\}_{[1]}$,$\{I2,I1\}_{[1]}$.

There are two nodes I2, I4 on the other branch, and the frequency for I4 =2, so we get the composition of $\{I4, I2\}_{[2]}$. All of these combinations are going to be a part of a superset Can_F.

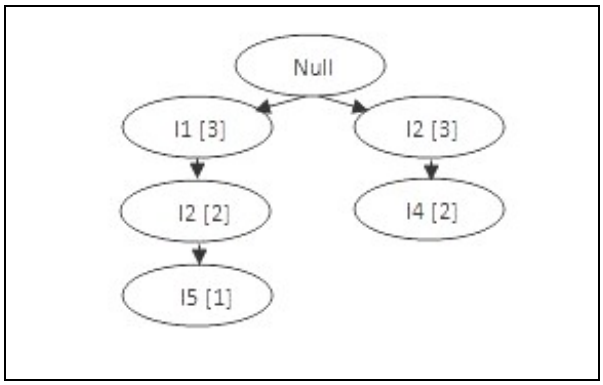

Figure 1: The combinations in the branch

The process of association rules mining consists of two phases:

Phase 1: Identify all frequent items by using Definition 2.

Phase 2: Extract rules of interest from these itemsets.

In phase 1which is to identify all the frequent itemset of length 1, first we need to read the database to construct the table of information (see Table 2).

Table 2*: The table of information*

Item	Frequency
I2	7
I1	6
I3	6
I4	2
I5	2
Number of Records	
RN	9

This table includes all the frequent 1-itemsets in descending order. In addition to the frequency of the items, the table contains the number of the rows RN.

Let's suppose that the value of NF in (Definition 3) equals four, the database will be divided into 4 folds. And the T900 will be added to fold 4. The purpose of dividing the database into 4 folds is to make it possible to fit in the memory, therefore the process of mining the frequent itemset is becoming faster.

For each fold ($fold \subseteq D$) a Prefixfoldtree is constructed, which retains the itemset association information. Figure 2 illustrates the process of converting the first fold to the Prefixfoldtree. After mining this Prefixfoldtree the tree will be deleted from the memory so the new Prefixfoldtree from the next fold will take its place.

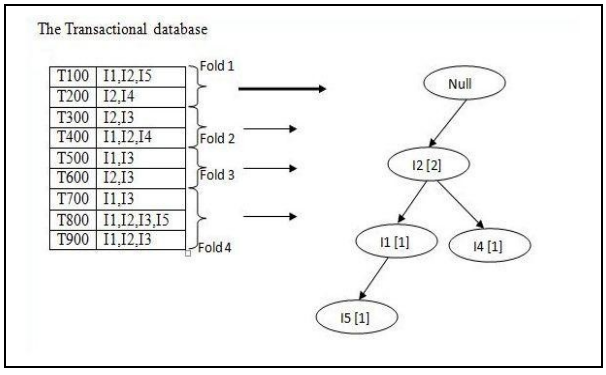

Figure 2: The Process of converting Fold 1 to Prefixfoldtree

After constructing the Prefixfoldtree a TOP (table of pointers) is a list of pointers is created. Every pointer in the TOP only points to a leaf node on the Prefixfoldtree and every pointer in the leaf is pointing only by a pointer. By using the TOP, the mining process of the Prefixfoldtree can be performed in the following steps:

1- The first pointer in the TOP points to the leaf node. The first pointer will retrieve the leaf node in the Prefixfoldtree.

2- Taking all the combinations that are composed of the nodes in that branch, not including the root.

3-Then adding labels and values to all the combinations (ex. $\{I5, I1\}_{[3]}$).

4- All of these combinations that are obtained from the Prefixfoldtree will be sent to the Can_F. If there is no such combinations in the Can_F then add them immediately to the set otherwise add the frequency of the new combination to the one of the existing combination.

5-Finally we modify the nodes in the branch by subtracting the leaf node's frequency from every other node's frequency value on the branch. Then if the base node of the leaf is not the root and does not have other sub-nodes or sub-node, we make the pointer in the TOP points to the base's node of the leaf or else delete the pointer in the TOP. In that way mining all the frequent itemsets in the Prefixfoldtree is done. For the first Prefixfoldtrees the TOP and the leaf nodes are shown in Figure 3.

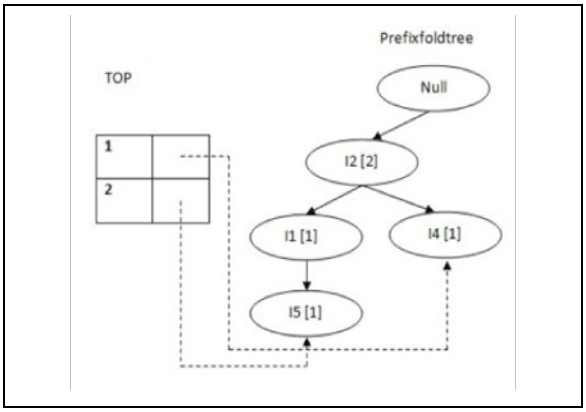

Figure 3: TOP table for the first fold

For the purpose of illustration, an example with the first Prefixfoldtree is illustrated. Let the min_sup =2.

In Figure 3 the first pointer in the TOP points to I4, and the branch node I4, I2 is obtained. Combine the two nodes (items) to get the combination $\{I4, I2\}_{[1]}$ then subtract the number of the I4, I2 from other nodes frequency on the branch, because the base node of I4 has other sub-nodes

which we do not delete. The second pointer in the TOP points to I5, The combination that is obtained is $\{I5, I1\}_{[1]}$, $\{I5, I2\}_{[1]}$,$\{I2,I1\}_{[1]}$,$\{I5, I1, I2\}_{[1]}$ because the set Can_F

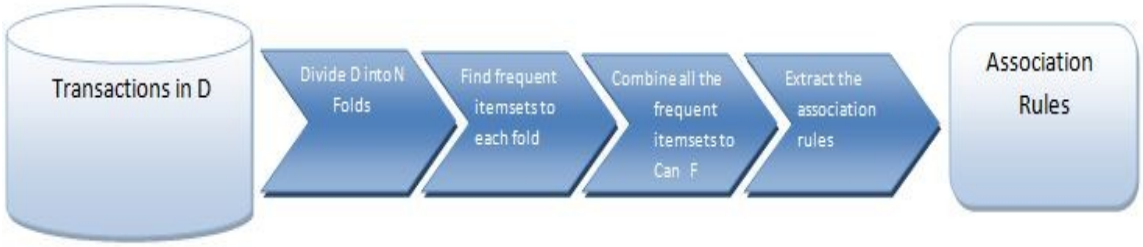

Figure 5: The PFT scheme

is empty at this moment; the two enter the Can_F set directly. The trees are deleted as soon as they are mined. For the second tree in the Figure 4, the first pointer in the TOP points to I3, We get the branch I3, I2 and combine the two nodes (items) to get the combination $\{I3,I2\}_{[1]}$. Then subtract the number of the I3, I2 from other nodes frequency on the branch, because the base node of I2 has other sub-nodes which we do not delete.

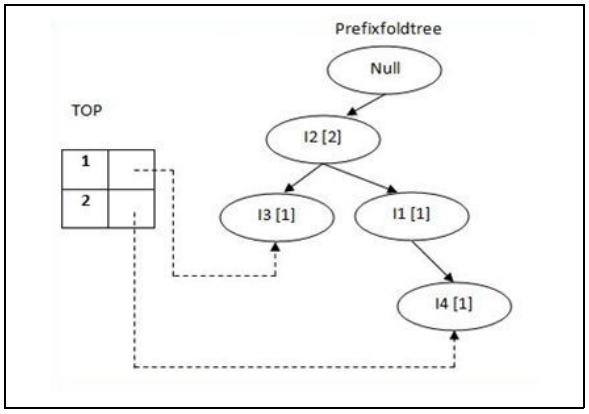

Figure 4: TOP table for the first fold

The second pointer in the TOP points to I4, The combination that we get is $\{I4,I1\}_{[1]}$,$\{I4,I2\}_{[1]}$,$\{I1,I2\}_{[1]}$,$\{I4, I1, I2\}_{[1]}$. At this point we check if there are similar combinations in Can_F set to those that were obtained from the second Prefixfoldtree. If there are similar combinations, then we add the frequency of the new combination to the existing one, or else we add it directly to the Can_F set. For the third and forth Prefixfoldtrees the same process will be repeated until we get the whole Can_F set.

Those combinations in set Can_F whose frequency is less than min_sup (min_sup=2) are deleted. Eventually, the superset (Can_F) will contain all the frequent itemsets ; Can_F=$\{\{I3,I1\}_{[4]}, \{I2,I1\}_{[4]}, \{I3,I2\}_{[4]},\{I5,I2\}_{[2]},\{I5,I1\}_{[2]}, \{I4,I2\}_{[2]},\{I5,I2,I1\}_{[2]},\{I3,I2,I1\}_{[2]}\}$.

3.3 The Measurements For The Rules

Since there are enormous numbers of frequent itemsets in a Can_F and only some of them are valuable for extracting association rules of interest, a pruning strategy is crucial to efficiently search for interesting itemsets. (Wu et al., 2004) presented pruning strategy which will be adapted to the purpose of searching the interesting itemsets. Itemsets of possible interest are captured by exploiting pruning function interest(X,Y). After identifying all frequent items in ICan_F, we extract rules of interest from these itemsets. The association rules of strong interest are extracted by using interestingness constraint functions correlation (X,Y) and CPIR(X/Y).The correlation measurement is used for finding the interesting relationships between data itemsets based on the correlation. While CPIR(X/Y) measurement is employed as the confidence measure of an association rules between itemsets X and Y.

In Figure 5, the main stages in the PFT scheme are illustrated

3.4 The Proposed Scheme

This is the pseudo-code of the Prefixfoldtree scheme:

Scheme: PFT

Input:

D: Database: NF; number of folds: min_sup; minimum support: min_conf; minimum confidence: min_int; minimum interest

Output: Association Rules

(1) **Begin**

(2) //*phase 1: reading the database and dividing it

(3) Fold← {the number of the transactions in the fold}

(4) **While** (! EOF)

(5) { RNcount++;}

(6) **If** (RN is odd) Fold= RN-1/NF;

(7) **Else** Fold=RN/NF;

(8) //*phase 2: construct the Prefixfoldtree for each fold.

(9) // This is done by using MFP algorithm

(10) MFP (Fold, min_sup);

(11) //*phase 3: finding the interesting itemsets

(12) **For** each K-itemset I ∈ Can_F **do**

(13) **If** I.count≥1 **then**

(14) **For** ∀ itemsets X,Y, X∪Y=I and X∩Y=∅ **do**

(15) **If** interest (X,Y)> min_int **then** ICan_F← {I}

(16) **End**

(17) **End**

(18) //*phase 4: extract the association rules AR from the ICan_F

(19) **For** each frequent k-itemset in the ICan_F **do**

(20) **If** correlation(X,Y)>1&& CPIR(Y|X)≥min_conf

(21) **then** AR← {X→Y}

(22) **If** correlation(X,Y)>1&& CPIR(X|Y)≥min_conf

(23) **then** AR← {Y→Y}

(24) **Return** Association rules

(25) **End**

With the aim of solving memory limitation and improving the quality of the association rules, in this paper we present a new scheme called PFT scheme for mining association rules from frequent patterns. This scheme adapts hybrid approach to deal with a large-scale of dataset. The original dataset is divided into folds, and then from each fold the frequent patterns are mined by using the tree projection approach and finally these frequent patterns are combined into one set. Interestingness constraints are used to assure the quality of the obtained rules. The next step is going to activity our approach and we do anticipate that our result is going to be better than Fpgrowth and Apriori algorithms in term of large-scale dataset.

5.0 REFERENCES

Aggelis, V. (2004) .”Association rules model of e-banking services”, In 5th International Conference on Data Mining, Text Mining and their Business Applications.

Agrawal, R. , Imieliński , T. , & Swami , A. (1993). ”Mining association rules between sets of items in large databases”.In proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, pages 207-216, Washington, DC.

Agrawal, R. , & Srikant, R. (1994) .“Fast algorithm for mining Association Rules”, In Proc. of the 1994 Int. Conf. Very Large Data Bases, Pages 487-499,Santiago, Chile.

Carlos, O. , Norberto, E. , & Santana ,C. A. (2006) . ”Constraining and summarizing association rules in medical data” Knowledge and information systems ISSN 0219-1377.

Encheva, S. , & Tumin, S. (2006). “Application of Association Rules for Efficient Learning Work-Flow“, IFIP International Federation for Information Processing, Vol. 228, Intelligent Information Processing III.pp. 499-504 .

Gao, J. (2007). “Realization of new Association Rule Mining Algorithm” Int. Conf. on Computational Intelligence and Security, IEEE.

Grahne, G. , & Zhu, J. (2005) . ”Fast Algorithm for frequent Itemset Mining Using FP-Trees”,IEEE Transactions on Knowledge and Data Engineer,Vol.17,NO.10.

Han, J. , Pei, J. , & Yin, Y. (2000) . “Mining frequent patterns without candidate generation” . In Proc. ACM-SIGMOD Int. Conf. Management of Data (SIGMOD ’96), Page 205-216.

Han, J. , & Kamber, M. (2006) . ”Data Mining : concepts and techniques” , second edition, The Morgan Kaufmann Series in Data Management Systems.

Liu, G. , Lu ,H. , Yu ,J. X., Wang, W., & Xiao, X. (2003) . ”AFOPT:An Efficient Implementation of Pattern Growth Approach”, In Proc. IEEE ICDM'03 Workshop FIMI'03.

Mannila, H. , Toivonen , H. , & Verkamo ,A. I. (1994). “Efficient Algorithms for Discovering Association Rules”, Proceedings of the AAAI Workshop on Knowledge Discovery in Databases, Usama M. Fayyad and Ramasamy Uthurusamy (Eds.), Washington, pp. 181-192.

Nguyen, S. N., Orlowska, M. E. , & Li ,X. (2008) . “Graph Mining based on a Data Partitioning Approach” .In proceedings 19th Australasian Database Conference ,Wollongong ,Australia.

Savasere, A. , Omiecinski, E. , & Navathe, S. (1995) . “An Efficient Algorithm for Mining Association Rules in Large Databases ”.In Proceedings 21st VLDB Conference ,Zurich ,Swizerland.

Srikant, R., & Agrawal , R. (1995) . “Mining Generalized Association Rules”, Proc. of the 21st VLDB Conference, Zurich, Switzerland.

Wang, J., Hsu ,W., Lee, M., & Sheng, C. (2006). “A partition-based Approach to Graph Mining” IEEE International Conference on Data Engineering (ICDE06).

Wu, X. , Zhang ,C. , & Zhang S. (2004) . ”Efficient mining of both positive and negative association rules” ACM Transactions on information Systems, pp.381-405.