An Improved Algorithm for Mining Frequent Inter-Transaction Patterns

Thanh-Ngo Nguyen
Faculty of Computer Science and Management,
Wroclaw University of Science and Technology, Wroclaw, Poland
ngont2011@gmail.com

Loan T.T. Nguyen
Faculty of Information Technology, Nguyen Tat Thanh University
Ho Chi Minh City, Vietnam;
Faculty of Mathematics Informatics and Mechanics,
University of Warsaw,
Warsaw, Poland
nthithuyloan@gmail.com

Ngoc-Thanh Nguyen
Faculty of Computer Science and Management,
Wroclaw University of Science and Technology, Wroclaw, Poland
ngoc-thanh.nguyen@pwr.edu.pl

Abstract-Mining Inter-transaction patterns (ITPs) from large databases is a common data mining task, which discovers the patterns across several transactions in a transaction database. Although, several algorithms have been proposed for this task, they remain computationally expensive. To resolve this issue, this paper presents an efficient method called DITP-Miner to mine ITPs. In our proposed algorithm, there are four phases. First, we scan the database once to find frequent 1-patterns with their tidsets. Second, we generate inter-transaction 1-pattern candidates with the given span values and sort all the frequent 1-patterns in an ascending order according to their supports. Third, based on frequent items found in phases 1 and 2, we find frequent 2-patterns with their diffsets. In the fourth phase, we use diffsets and DFS (Depth-First-Search) technique to get all frequent ITPs. In addition, three propositions are also offered to early prune infrequent patterns in the processing. Proposition 1 is used to early prune infrequent inter-transaction 1-pattrens, Proposition 2 is used to quickly compute the support of patterns, and Proposition 3 (subsume concept) is used to quickly compute the support of patterns and to early prune infrequent patterns, which reduce the search space. Through experimental results, we find out our proposed approach is more efficient than ITP-miner in both the mining time and the memory usage.

Keywords-Data mining, inter-transaction association rules, inter-transaction patterns, inter-transaction itemsets.

I. INTRODUCTION

Association rule is one of the most essential tools for predictive science. Association rule mining is also a common task and plays an important role in the field of data mining. Mining association rules from large databases has emerged as a very exciting research topic in recent years. Like Apriori [1-3] approaches, candidate generation methods using a level-wise approach, have fairly large processing time because of scanning database many times. The FP-Growth [4] and FP-Growth* [5] approaches, divide-and-conquer methods which compress the database into a tree structure, require much time to travel of FP-tree to mine frequent itemsets (FIs). Methods with vertical data format to compress the database and mine frequent itemsets use a divide-and-conquer strategy. DBV-FI [6], Index-BitTableFI [7], Eclat [8], dEclat [9] and, constraint [10] are some examples. The Eclat algorithm based on IT-tree needs to scan the databases only once for forming tidsets of items. Therefore, the runtime is sharply reduced, and this is a better method. However, the main drawback of Eclat algorithm is that it uses much memory to store tidsets. This leads to some difficulties in calculating the intersection of tidsets between two itemsets in the equivalence class, especiallyfor large databases with the number of transactions up to millions.

The primitive association rule methods find out association rules among itemsets within a transaction (intra-transaction). The tradition methods only can predict the rule "If the stock prices of Apple and Samsung go up, the price of Intel can go up on the same day". These approaches cannot be applied for the form of rule "If the stock prices of Apple and Samsung go up, the price of Intel can go up several days later". These rules show that association relationships among the itemsets span different transactions. Therefore, we call these rules as inter-transaction association rules (ITARs).

Many methods have been done for mining ITARs in recent years [11-16]. These methods used vertical data structure (like ECLAT algorithm) and an ITP-tree to mine all frequent ITPs in a depth-first search manner. It has been revealed that ITP-Miner outperforms previous ITPs mining algorithms. However, ITP-Miner algorithm still uses tidsets to store nodes of frequent ITPs, thus ITP-Miner consumes much memory usage and execution time. Wang et al. proposed a approach, which is bsaed on projections, called the PITP-Miner [17] algorithm for efficient mining of frequent ITPs in a large database. Moreover, Hsieh et al. proposed the PRMiner [18] algorithm to mine profit rules for the interday trading model, then Hsieh et al. develped two algorithms of JCMiner and ATMiner[19] algorithms to mine profit rules from closed ITPs.

In our method, we propose an improved algorithm using diffsets [9], called DITP-Miner, for fast mining ITPs from transaction databases. We present an algorithm for storing and processing tidsets of an itemset to quickly mine frequent ITPs using DITP-tree. This method shows the effectiveness in mining frequent ITPs through experiments.

The remains of this paper is organized as follows. Section II presents concepts and some definitions related to mining frequent ITPs. Section III reviews some related works, Section IV presents the data structure of DITP-tree, DITP-Miner algorithm and illustrative examples. Section V discusses our experimental results of applying DITP-Miner and the conclusions and future works are discussed in Section VI.

II. BASIC CONCEPTS

In this section, the notations and basic definitions will be introduced to describe problems of mining frequent ITPs.

TABLE I. BASIC NOTATIONS

D	Transaction database
ms	Minimum support threshold
msp	Maximum span value threshold
X	An itemset or a pattern
w	Span value
k-pattern	A pattern with k items
$\sigma(X)$	Support of a pattern X
d(XY)	Diffset of pattern X and Y
Tidset	A set of domain attributes
[P]	A equivalence class at P
Span	A given value so that $0 \le Span \le msp$

Definition 1. [22] Let $I = \{i_1, i_2, \ldots, i_m\}$ be a set of distinct items, and $A = \{a_1, a_2, \ldots, a_n\}$ be a set of transaction identifications. A transaction database *(D)* contains a set of transactions in the form of $\langle tid, T_{tid}\rangle$, where $tid \in A$, $T_{tid} \subseteq I$, T_{tid} is an itemset, and *tid* is the transaction identification of T_{tid} that describes a property, such as the time stamp associated with T_{tid}. We can also say that itemset T_{tid} occurs in a transaction with the transaction identification *tid*, or occurs at *tid*.

Definition 2. [22] Let $\langle x, T_x\rangle$ and $\langle y, T_y\rangle$ be two transactions in*D*. The relation of distance between *x* and *y* is defined as $(x - y)$, where $x > y$, and *y* is called the reference point. Regarding to *x,* an item i_k at *x* is called an extended item and denoted as $i_k(x - y)$, where $(x - y)$ is called the span of the extended item. Similarly, regarding to *y*, a transaction T_x at *x* is called an extended transaction and denoted as $T_x(x - y)$. Thus, an extended transaction consists of a set of extended items, i.e., $T_x(x - y) = \{i_1(x - y), \ldots, i_s(x - y)\}$, where *s* is the number of items in T_x.

Table II. describes *D*, the extended transaction of the transaction at *tid* = 2 with respect to the transaction at *tid* = 1 is $\{a(2), b(2)\}$, when we consider in the inter-transaction context $\{a(2), b(2)\}$ similar to $\{a(0), b(0)\}$.

Definition 3. [16] Let $u_i(l_i)$ and $u_j(l_j)$ be two extended items. $u_i(l_i) < u_j(l_j)$ if (1) $l_i < l_j$, or (2) $l_i = l_j$ and $u_i < u_j$. Moreover, $u_i(l_i) = u_j(l_j)$ if $l_i = l_j$ and $u_i = u_j$.
For example, $a(0) < b(0)$, and $b(0) < a(1)$.

TABLE II. A EXAMPLE OF TRANSACTION DATABASE *(D)*

Tid	Items
1	{*a, b*}
2	{*a, c, d*}
3	{*a*}
4	{*a, b, c, d*}
5	{*a, b, d*}
6	{*a, d*}

Definition 4. An inter-transaction pattern is defined as a set of extended items, $\{n_1(l_1), n_2(l_2), \ldots, n_k(l_k)\}$, where $l_1 = 0$, $l_k \le msp$, *msp* is a maximum *Span*, $n_i(l_i) < n_j(l_j)$, and $1 \le i < j \le k$.

Definition 5. [22] A given transaction $\langle d_1, T_{d1}\rangle$ in *D*, a mega-transaction, M_{d1}, is defined as a set of extended transactions in *D* regarding to d_1, i.e., $M_{d1} = T_{d1}(0) \cup T_{d2}(d_2 - d_1) \cup T_{d3}(d_3 - d_1) \cup \ldots \cup T_{dk}(d_k - d_1)$, where $\langle d_1, T_{d1}\rangle$, $\langle d_2, T_{d2}\rangle, \ldots, \langle d_k, T_{dk}\rangle$ are consecutive transaction in *D,* $d_i < d_{i+1}$, $i<k$, and $d_k - d_1 \le msp$. Notice that M_{d1} is also an inter-transaction itemset, where d_1 is the reference point.

Definition 6. The cardinality of extended items in a pattern is called the length of the pattern. A pattern of length *l* is called a *l*-pattern.

For example, $\{b(1), d(1)\}$ and $\{a(0), a(1), b(1), c(1), d(1)\}$ are 2-pattern and 5-pattern, respectively.

Definition 7. Let $P=\{p_1(i_1), p_2(i_2), \ldots, p_m(i_m)\}$ and $Q=\{q_1(j_1), q_2(j_2), \ldots, q_n(j_n)\}$ be two ITPs, respectively. We say that $P = Q$ if $p_i(i_i) = q_i(i_j)$ for $1 \le i \le m$, where $i_1=j_1=0$ and $m=n$. We also say that $P<Q$, if (1) $p_1(0) < q_1(0)$; or (2) there exists *k* (≥ 1) such that $p_i(i_i) = q_i(j_i)$ for $1 \le i \le k$, and $p_{k+1}(i_{k+1}) < q_{k+1}(j_{k+1})$.

For example, $\{b(0), d(1)\} = \{b(0), d(1)\}$, and $\{a(0), b(1), d(1)\} < \{a(0), d(1)\}$.

Definition 8. Given *D, ms*, *Y* be an inter-transaction itemset. The support of *Y*, *sup*(*Y*), is the munber of records in *D* contains Y. If $sup(Y) \ge ms$, then *Y* is a frequent inter-transaction pattern.

III. RELATED WORKS

In 2007, Lee et al. proposed the ITP-Miner [16] algorithm to mine frequent ITPs, the ITP-Miner algorithm using the ECLAT [8] algorithm and the DFS (Depth First Search) strategy to traverse the ITP-tree to get all frequent ITPs. In 2008, Lee et al. also proposed the ICMiner [22] algorithm to mine closed ITPs in transaction databases, the ICMiner used the CHARM [8] algorithm, applied the properties of frequent closed patterns, and DFS strategy to travel the ICD-Tree to find all common closed patterns. In 2011, Wang et al. proposed an algorithm called PITP-Miner [17] using a projection-based approach to discover all frequent ITPs. In 2014, The authors developed the PRMiner [18] approach to mine profit rules on the large stock databases and then in 2016, these authors developed two methods of JCMiner and ATMiner [19] processing closed itemsets, which outperforms the PRMiner approach, especially for databases with the large number of items. In 2015, Wang developed methods to mine non-redundant inter-transaction rules. Wang developed the NRITR-Miner [20] algorithm to mine the non-redundant inter-transaction rules efficiently and the ITR-Miner [20] algorithm to mine the full-set of inter-transaction rules.

The above methods exploit relatively well the common set of common transactions. However, these methods use tidsets to store transaction information and the support of popular episodes. Therefore, they consume a lot of memory and execution time. In this work, we propose an improved DITP-Miner algorithm to efficiently exploit inter-generational distributions.

IV. MINING INTER-TRANSACTION PATTERNS

A. DITP-Miner algorithm

Definition 9. [22] Let $p = \{p_0(0), p_i(i_1),\ldots, p_{k-1}(i_{k-1})\}$ and $q = \{q_0(0), q_i(j_1),\ldots, q_{k-1}(j_{k-1})\}$ be two frequent k-patterns in the equivalence class $[P]$, $k > 1$, p and q can combine to form a frequent $(k+1)$-pattern if the first $(k-1)$ extended items of p are equal to those of q, and the last extended item of p is less than that q. The joined pattern is got by appending the last extended item of q to p; the joined pattern is $\{ p_0(0), p_i(i_1),\ldots, p_{k-1}(i_{k-1}), q_{k-1}(j_{k-1})\}$.

Definition 10. [8] Diffset of itemset X and Y of an equivalence class, $d(XY)$, is defined as follows

- $d(PX) = t(P) - t(X) \Longrightarrow \sigma(PX)= \sigma(PX) - |d(\mathrm{PXY})|$
- $d(PXY)=d(PY) - d(PX)$

Diffsets are often smaller than tidsets. Therefore, using diffsets to count the support that reduces the memory usage of storing tidsets when we mine frequent patterns from large databases with a large amount of records.

Proposition 1. Let $u(0)$ and set of integers $\{t_1, t_2, \ldots, t_n\}$ be a frequent 1-pattern and its tidset, respectively. With a given positive integer w, $1 \leq w \leq msp$, we can generate ITPs $u(w)$ with its tidset $\{t_1\text{-}w, t_2\text{-}w, \ldots, t_n\text{-}w\}$ in which $t_i - w \geq 1$. We can prune $\{u(w)\}<t_1\text{-}w, t_2\text{-}w, \ldots, t_n\text{-}w>\}$ ITPs if the support of these ITPs is less than ms due to *downward closure property* of the Apriori principle. This can help DITP-Miner algorithm accelerate the performance and save memory usage at the length-1 level without computation of the hash table $H2$ used in ITP-Miner algorithm.

Proposition 2. If the length of a tidset in a *k-pattern* minus a *Span* whose value varies from 1 to *msp* is smaller than the *ms*, we do not need to generate (k+1)-patterns responding to the *Span*.

Proposition 3. Suppose that X, Y are two frequent patterns. If X's tidset is the subset of Y's tidset, then pattern X is subsumed in pattern Y. This means that the support of pattern XY is equal to that of pattern X ($Sup(X) = Sup(XY)$).

Input: D, ms, msp.
Output: All frequent ITPs *FPs.*
Method:

1 Scan D to find all frequent 1-patterns α with $Span=0$ and their tidsets
2 Generate frequent inter-transaction 1-patterns γ with the given *Span* values w ($1 \leq w \leq msp$) by applying the proposed propositions.
3 Insert the frequent 1-patterns α into the DITP-tree T.
4 **for each** frequent 1-pattern α **do**
5 **for each** frequent 1-pattern γ **do**
invoke Join_2($\alpha,\gamma,T, ms, msp,|D|$) to find all frequent inter-transaction 2-patterns β;
6 **for each** frequent inter-transaction 2-pattern θ in β **do** invoke DFS($\theta,FP,T, ms, msp,|D|$);
7 **end for**
8 Output *FPs.*

Fig. 1. The DITP-Miner algorithm

The procedure Join_2 is used to generate frequent inter-transaction 2-patterns while the procedure DFS is used to recursively traverse the DITP-Tree to generate all frequent ITPs.

B. Examples

Let us consider the example as shown in Table 2, where $ms = 3$, and $msp = 1$. We describe how to join $\{a(0),d(0)\}$ to every pattern in the equivalence class of $\{a(0)\}$ to find frequent 3-patterns.

For example, let us consider $\{a(0),b(0)\}<1,4,5>$ and $\{a(0)d(0)\}<2,4,5,6>$

$d(\{a(0),b(0)\},\{a(0),d(0)\})=t(\{a(0),b(0)\})-t(a(0),d(0)\})=\{1,4,5\}\backslash\{2,4,5,6\}=\{1\}$
$\Rightarrow Sup(\{a(0),b(0)\},\{a(0),d(0)\})=Sup(\{a(0),b(0)\} - d(\{a(0),b(0)\},\{a(0),d(0)\}) = 3\text{-}1 = 2 < ms = 3$
Therefore, $\{a(0),b(0),d(0)\}$ is pruned.

Let us consider $\{a(0),b(0)\}$ $<1,4,5>$ and $\{a(0)a(1)\}$ $<1,2,3,4,5>$

$d(\{a(0),b(0)\},\{a(0)a(1)\})=t(\{a(0)b(0)\})-t(a(0),a(1)\})=\{1,4,5\}\backslash\{1,2,3,4,5\}=\{\emptyset\}$
$\Rightarrow Sup(\{a(0),b(0)\},\{a(0),a(1)\})= \sigma(\{a(0),b(0)\} - d(\{a(0),b(0)\},\{a(0),a(1)\}) = 3\text{-}0 = 3 = ms$
Therefore, $\{a(0),b(0),a(1)\}$ is added to ITPDIFF-tree.

The other k-patterns can be obtained from the $(k\text{-}1)$-patterns in the same way. Besides, we can use "*Downward Closure Property*" to prune k-patterns.

The frequent ITPs and their diffsets generated during the mining processes are shown in Fig. 2.

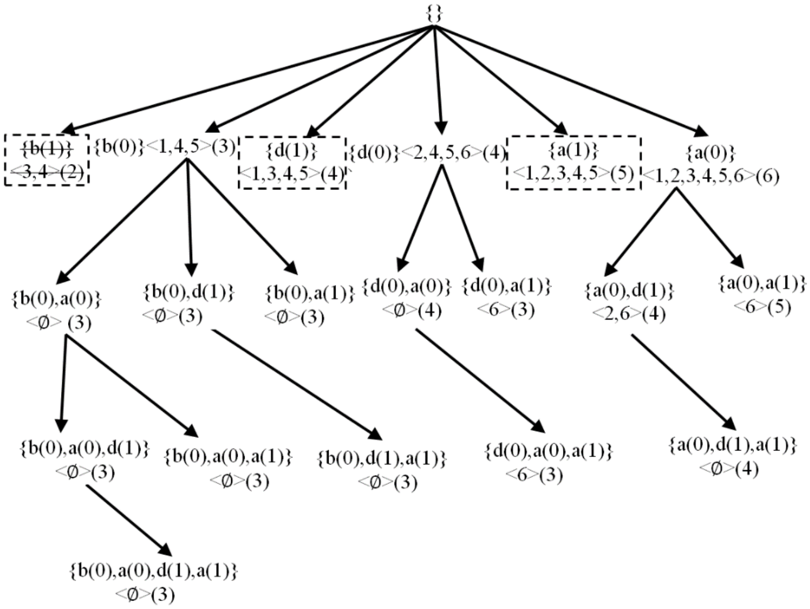

Fig. 2. The DITP-tree of the improved algorithm, DITP-Miner for the database shown in Table II.

V. Experimental Results

A. Characteristics of experimental databases

The experimental algorithms were implemented with MS Visual CSharp 2013, and performed on a PC with Windows 10 OS, CPU Intel(R) Core(TM) i5-5287U, 2.90 GHz, and 8.00 GB RAM.

Databases used for experiment downloaded from UCI Machine Learning Repository (http://mlearn.ics.uci.edu) and their charesteristics shown in Table III.

TABLE III. CHARACTERISTICS OF EXPERIMENTAL DATASETS

#Dataset	#distinct items	#records
Chess	76	3,196
Accidents	468	340,183
Connect	120	8,124

1) Execution time:

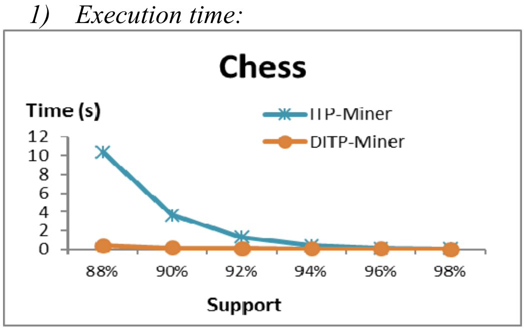

Fig. 3. Execution time of ITP-Miner and DITP-Miner for Chess dataset (Support).

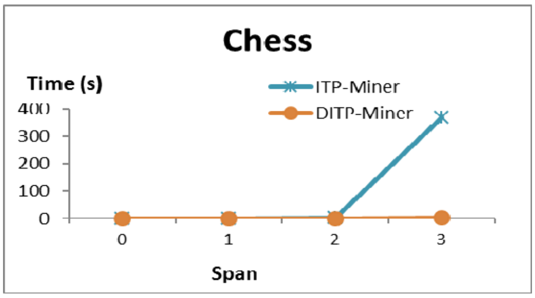

Fig. 4. Execution time of ITP-Miner and DITP-Miner for Chess dataset (Span).

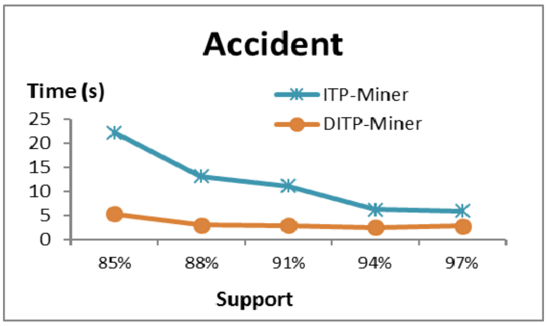

Fig. 5. Execution time of ITP-Miner and DITP-Miner for Accidents dataset (Support).

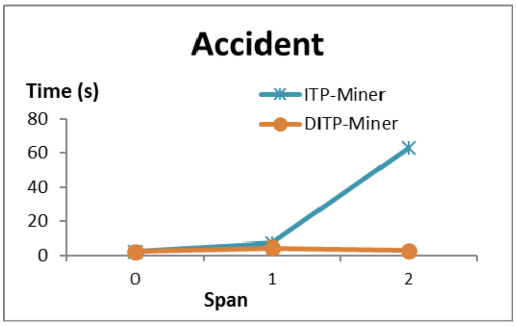

Fig. 6. Execution time of ITP-Miner and DITP-Miner for Accidents dataset (Span).

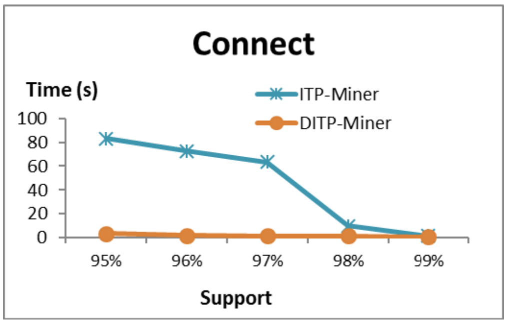

Fig. 7. Execution time of ITP-Miner and DITP-Miner for Connect dataset (Support).

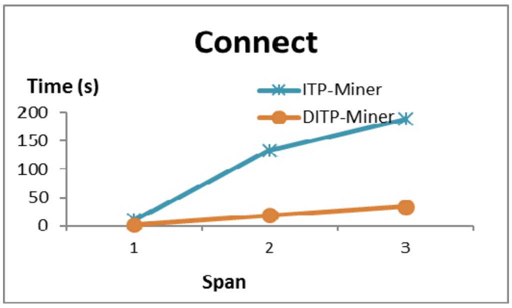

Fig. 8. Execution time of ITP-Miner and DITP-Miner for Connect dataset (Span).

2) Memory usage:

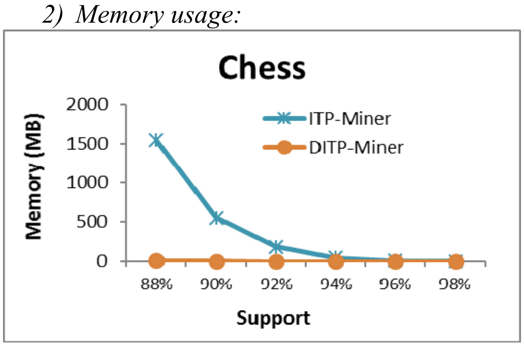

Fig. 9. Memory usage of ITP-Miner and DITP-Miner for Chess dataset (Support).

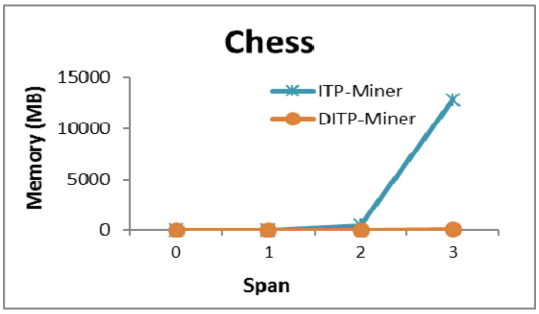

Fig. 10. Memory usage of ITP-Miner and DITP-Miner for Chess dataset (Span).

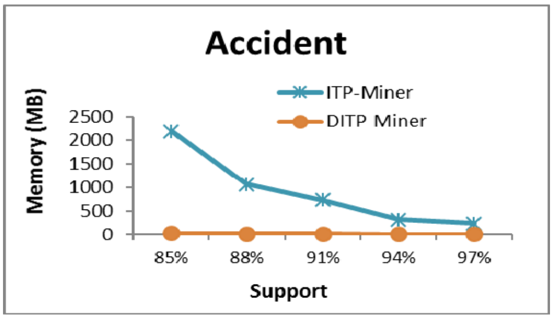

Fig. 11. Memory usage of ITP-Miner and DITP-Miner for Accident dataset (Support).

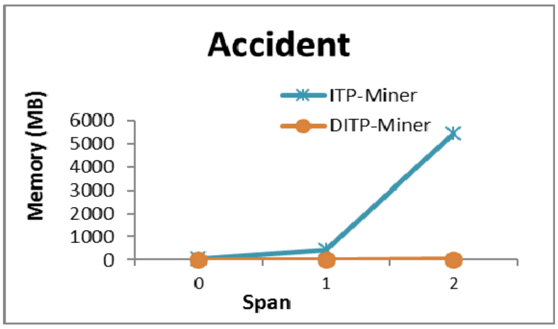

Fig. 12. Memory usage of ITP-Miner and DITP-Miner for Accidents dataset (Span).

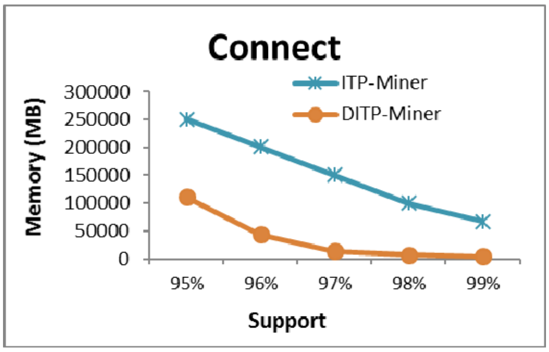

Fig. 13. Memory usage of ITP-Miner and DITP-Miner for Connect dataset (Support).

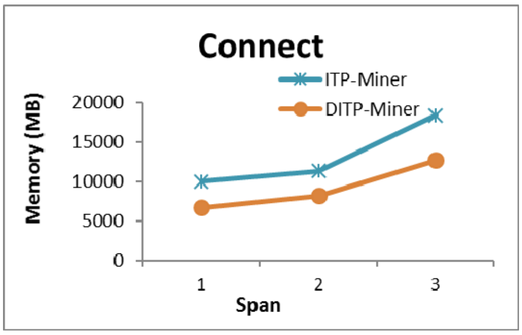

Fig. 14. Memory usage of ITP-Miner and DITP-Miner for Connect dataset (Span).

Experiential results shown in Figures 3-14 demonstrate that our proposed algorithm outperforms the ITP-Miner algorithm in saving processing time and memory. Therefore, our proposed approach can be considered to be a significant contribution to mining frequent ITPs, which is a task important to generate association rules.

VI. Conclusion and Future Works

This paper has proposed an effective approach using diffsets to reduce the memory usage for storing tidset information, generate 1-patterns with given span values and offer three properties used to early prune infrequent patterns at length-1 level, to ignore the computation of the support in some special cases, and to quickly compute the support in mining frequent inter-transaction from large transaction databases. The experiments executed on three databases show that our proposed approach outperforms the current approach, ITP-Miner, in terms of memory usage and runtime. The proposed algorithm can be used, among others, in determining multiple classifier-based spatiotemporal features [23].

In the future, we will further study in the scope of mining FITPs using diffsets with hierarchical databases. Besides, we will apply this method to mine frequent closed ITPs, maximal frequent ITPs.

Acknowledgment

This work was carried out during the tenure of an ERCIM ‘Alain Bensoussan’ Fellowship Programme.

References

[1] R. Agrawal, T. Imielinski, A. Swami, Mining association rules between sets of items in large databases, In Proc. of ACM SIGMOD’93, pp. 207–216, 1993.

[2] R. Agrawal, R. Srikant, Fast algorithms for mining association rules, In Proc. of VLDB’94, pp. 487–499, 1994.

[3] R.J. Bayardo, Efficiently mining long patterns from databases, In Proc. of ACM SIGMOD, pp. 85–93, 1998.

[4] J. Han, J. Pei, Y. Yin, R. Mao, Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach, Data Min. Knowl. Discov. 8(1): 53-87, 2004.

[5] G. Grahne, J. Zhu, Fast Algorithms for Frequent Itemset Mining Using FP-Trees.IEEE Transactions on Knowledge And Data Engineering, 17(10), pp.1347-1362, 2005.

[6] J. Dong, M. Han, BitTableFI: An efficient mining frequent itemsets algorithm, Knowledge-Based Systems, 20, pp. 329-335, 2007.

[7] W. Song, B. Yang, Z. Xu, Index-BitTableFI: an improved algorithm for mining frequent itemsets, Knowl Based Syst 21:507–513, 2008.

[8] M.J. Zaki, C.J. Hsiao, Efficient algorithms for mining closed itemsets and their lattice structure, IEEE Transactions on Knowledge and Data Engineering 17 (4), pp. 462–478, 2005.

[9] M.J. Zaki, K. Gouda, Fast vertical mining using diffsets, in: Proceedings of the 9th ACM-SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, pp. 326–335, 2003.

[10] H.V. Duong, T.C. Truong, An efficient method for mining association rules based on minimum single constraints. Vietnam Journal of Computer Science, 2(2), pp. 67-83, 2015.

[11] L. Feng, T.S. Dillon, J. Liu, Inter-transactional association rules for multi-dimensional contests for prediction and their application to studying meteorological data, Data and Knowledge Engineering 37 (1), pp. 85–115, 2001.

[12] L. Feng, J.X. Yu, H. Lu, J. Han, A template model for multidimensional inter-transactional association rules, The VLDB Journal 11 (2), pp. 153–175, 2002.

[13] Q. Li, L. Feng, A. Wong, From intra-transaction to generalized inter-transaction: landscaping multidimensional contexts in association rule mining, Information Sciences 172, pp. 3–4, 2005.

[14] H. Lu, L. Feng, J. Han, Beyond intratransaction association analysis:mining multidimensional inter-transaction association rules, ACM Transactions on Information Systems, 18(4), 423–454, 2000.

[15] A.K.H. Tung, H. Lu, J. Han, L. Feng, Efficient mining of intertransaction association rules, IEEE Transactions on Knowledge and Data Engineering 15 (1), pp. 43–56, 2003.

[16] A.J.T. Lee, C.-S. Wang, An efficient algorithm for mining frequent inter-transaction patterns, Information Sciences 177, pp. 3453–3476, 2007.

[17] C.S. Wang, Kuo-Chung Chu, Using a projection-based approach to mine frequent inter-transaction patterns, Expert Systems with Application 38(9), pp. 11024-11031, 2011.

[18] Y. L. Hsieh, D. L. Yang, and J. Wu, Effective application of improved profit-mining algorithm for the interday trading model, The Scientific World Journal, Vol.2014, pp. 1-13, 2014.

[19] Y.L. Hsieh et al., Efficient Mining of Profit Rules from Closed Inter-Transaction Itemsets, Journal of Information Science and Engineering 32.3, pp. 575-595, 2016.

[20] C.-S. Wang, Mining Non-Redundant Inter-Transaction Rules, J. Inf. Sci. Eng. 31(6), pp. 1521-1536, 2015.

[21] J. Dong, M. Han, BitTableFI: An efficient mining frequent itemsets algorithm, Knowledge-Based Systems, 20, 329-335, 2007.

[22] A.J.T. Lee et al., An efficient algorithm for mining closed inter-transaction itemsets. Data & Knowledge Engineering 66, pp. 68-91, 2008.

[23] V.D. Hoang, Multiple classifier-based spatiotemporal features for living activity prediction. Journal of Information and Telecommunication, 1, pp. 100-112, 2017.