Association Rule Mining with IST-EFP Algorithm The Novelty of FP-Growth Algorithm

Boby Siswanto
Student
Dept of School of Science
Telkom University
Bandung, Indonesia
e-mail: boby.siswanto@gmail.com

Dr. The Houw Liong
Supervisor
Dept of School of Science
Telkom University
Bandung, Indonesia
e-mail: thehl007@gmail.com

Shaufiah, MT
Supervisor
Dept of School of Science
Telkom University
Bandung, Indonesia
e-mail: shaufiah@gmail.com

***Abstract*—EFP algorithm is an FP-Growth algorithm that is applied to the DBMS such as Oracle DBMS. IST-EFP algorithm is the improvement of EFP algorithm which is combined by intersection of set theory analysis. IST-EFP algorithm will reduce items from the original dataset that have low support value. IST-EFP algorithm has better performance than EFP algorithm because it reduces more items while maintaining the confidence values of the association rules obtained.**

Keywords—EFP algorithm; IST-EFP algorithm; Oracle DBMS; item reductions; maintain confidence values; association rules.

INTRODUCTION

Association rules are rules that relate one thing to another thing in which two things are closely interrelated. Association rules categorized as unsupervised learning because it can generate rules dynamically in terms of number and variety [1]. There are two reference values are calculated on association rules which are support values and confidence values. Support value is the ratio of one variable against frequency of the entire transaction. Support value will be compared with the value of the specified minimum support, only support values equal to or greater that will be processed further. Confidence value is a value that indicates the relationship of one variable with another ones, it compares the value of two or more variables combination. Result that will be taken from association rules are values that meet the minimum support and minimum confidence [2] [3]. These results can be used to determine managerial business actions

FP-Growth is an association rules algorithm which does not generate candidate itemset. FP-Growth forming tree structures by going through the stages of generation FP-Tree, Conditional Pattern Bases, Conditional FP-Trees and ends with the formation of Frequent Pattern [5]. Frequent pattern will generate rules that will generate confidence values. Stages of sufficient length requires a relatively large resource that is used depends on the size of the dataset. Many previous studies which seek to minimize the time to make the FP-Growth algorithm process more effective by create new and better methods [6] [7] [8] [9].

There is a previous research that introduces the use of tables in the DBMS to form FP-Tree. The reason of using the tables is to avoid long time of dataset processing in the main memory when forming the FP-Tree due to main memory capacity is not greater than the capacity of secondary storage. In this study, the FP-Tree algorithm using the object as a shaper table FP-Tree in the call to Expand Frequent Pattern (EFP) table.

Basically frequent pattern is a set of items that have relevance to transactions on the FP-Tree. Set theory is able to analyze related items in a set, which are intersection, union or subsets. A rule is a relationship between the correlated items. In DBMS, relation between tables is the application of set theory intersection. For example found table X = {a, b, c) is related to the table Y = {z, a, b} on attributes a and b, in theory, the set can be written as $X \cap Y = \{a, b\}$. The results of several sets of relationships will generate a new set with a smaller number of attributes. The dataset will consist of one or a few records and a record will be made up of several items. Record is a representation of the set and the item is a representation of the attribute. By using the intersection of set theory and unsupervised learning of FP-Growth algorithm would be produces new dataset with smaller on dimensions / sizes.

OBJECTIVES

Current research will apply a new method to improve EFP algorithm processing by reducing the size of the dataset (preprocessing phase). Confidence values from obtained rules will relatively unchanged.

Comparison results between original FP-Growth, EFP and IST-EFP will be evaluated.

FP-GROWTH ALGORITHM

FP-Growth algorithm will generate frequent patterns from a dataset. To gain the frequent pattern must go through the stages of (1) the ordering of items in descending order of frequency, (2) the formation of the FP-Tree, (3) Establishment of Conditional Pattern Bases, conditional FP-Tree and (4) Frequent Pattern Formation.

Here is an implementation of fp-growth algorithm to obtain the frequent pattern from the dataset:

TABLE I. A DATASET WITH MINIMUM SUPPORT TRESSHOLD = 20%

TID	Items	Sorted Frequent Items
c1	d,g,f,p,s	f,g,s,p
c2	q,i,l,n,j	i,l,q,j,n
c3	q,e,f,i,a	i,e,f,q,a
c4	s,e,f,q,t	e,f,q,s,t
c5	a,i,l,e,j	i,e,l,j,a
c6	f,p,j,h,l	l,f,j,p,h
c7	m,r,k,g,s	g,r,s,m
c8	h,i,l,t,m	i,l,h,t,m
c9	q,r,n,g,c	g,q,r,n
c10	e,r,b,g,i	i,e,g,r

Frequent Pattern obtained will be as follow:

TABLE II.
FREQUENT PATTERN WITH MINIMUM SUPPORT TRESSHOLD = 20%

Item	Freq. Pattern
r	g r : 3
l	j l : 3, i l : 3, h l : 2
i	e i : 3, a i : 2
j	i j : 2
f	e f : 2
e	a e : 2
q	i q : 2, n q : 2, e q : 2, f q : 2
p	f p : 2
s	g s : 2, f s : 2

EFP ALGORITHM

EFP algorithm will accelerate the frequent pattern formation process conducted by the FP-Growth algorithm. The dataset will be instantly transformed into EFP table where the EFP table is a representation of the FP-Tree.

TABLE III. AN EFP TABLE

TID	ITEM	PRV
c1	d	
c1	s	p
c1	p	g
c1	g	f
c1	f	d
c10	b	
c10	r	i
c10	i	g
c10	g	e
c10	e	b

TID	ITEM	PRV
c2	i	
c2	q	n
c2	n	l
c2	l	j
c2	j	i
c3	a	
c3	q	i
c3	i	f
c3	f	e
c3	e	a

TID	ITEM	PRV
c4	e	
c4	t	s
c4	s	q
c4	q	f
c4	f	e
c5	a	
c5	l	j
c5	j	i
c5	i	e
c5	e	a

TID	ITEM	PRV
c6	f	
c6	p	l
c6	l	j
c6	j	h
c6	h	f
c7	g	
c7	s	r
c7	r	m
c7	m	k
c7	k	g

TID	ITEM	PRV
c8	h	
c8	t	m
c8	m	l
c8	l	i
c8	i	h
c9	c	
c9	r	q
c9	q	n
c9	n	g
c9	g	c

An EFP table will be resides on DBMS. By using SQL aggregate function on equation 1 frequent pattern will be obtained with the results as on TABLE II.

$$\text{item, prv } \vartheta \text{ count(*) } (\Pi_{item,prv}(\sigma_{prv \text{ is not null}} (\text{EFP\_Table}))) \quad (1)$$

IST-EFP ALGORITHM

Set theory analysis is done by comparing the set of frequent pattern obtained by EFP algorithm compared to the set of datasets. The comparison is done slices / Intersection. The results of the analysis are taken is a dataset with a smaller size of the initial dataset will be obtained. The following algorithm IST-EFP (Intersection Set Theory - EFP) in pseudocode to obtain new datasets:

IST-EFP(Dataset, minSupCount)
1. X = Dataset
2. X_1 = CREATE temporary table FROM X WHERE COUNT(*) > minSupCount
3. Y_1 = CREATE EFP table FROM X_1
4. $Z = Y_1 \cap X$ on item WHERE Y_1.previtem is not null
5. Return Z

Figure 1: IST-EFP Algorithm

Suppose the data set used is from TABLE I, X1 = {i, e, q, h, k, d, s, j, g, r, p, b, a, t, c, f, n, l, m}. By applying the EFP algorithm then obtained a dataset with 11 items, Y1 = {i, e, q, h, j, g, a, l, m, n, f,}. Based on step number 4 in the IST-EFP algorithm then we will get a new dataset with 9 items, Z = {i, e, h, j, g, a, l, n, f}. Items that are removed are {k, d, s, r, p, b, t, c} where item {d, c, k, b} has a support count value 1 (automatically pruned) so pruning is done on items {p, r, s, t }.

TABLE IV.
A NEW DATASET AFTER PRUNED WITH IST-EFP ALGORITHM

TID	Items	Sorted Frequent Items
c1	~~d~~,g,f,~~p~~,~~s~~	f,g
c2	q,i,l,n,j	i,l,q,j,n
c3	q,e,f,i,a	i,e,f,q,a
c4	~~s~~,e,f,q,t	e,f,q
c5	a,i,l,e,j	i,e,l,j,a
c6	f,~~p~~,j,h,l	l,f,j,h
c7	m,r,~~k~~,g,~~s~~	g,r,m
c8	h,i,l,t,m	i,l,h,m
c9	q,r,n,g,~~e~~	g,q,n
c10	e,r,~~b~~,g,i	i,e,g

EVALUATIONS

Evaluation done by analyzing experiment results on three datasets. Dataset used will be named with the pattern M_Txx_Cxx code. Txx indicates the number of transactions/records and Cxx indicates the number of the attributes/items of the datasets.

TABLE V.
PERFORMANCE COMARISON BETWEEN EFP AND IST-EFP

No	Dataset	Number of Item			Time (s)		Number of Rules	
		Original	EFP	IST-EFP	EFP	IST-EFP	EFP	IST-EFP
1	M_T10_C344	344	51	43	0.93	0.63	137	110
2	M_T100_C1558	1,558	748	674	32.04	23.22	4,864	4,678
3	M_T1000_C3162	3,162	2,700	2,325	1,809.98	1,478.47	110,146	108,887

EFP algorithm on M_T10_C344 dataset able to shrink 85.17% number of items, while IST-EFP algorithm able to shrink 87.50%. On M_T100_344 dataset, EFP algorithm able to shrink 51.99% number of items and IST-EFP algorithm able to shrink 56.74%. On M_T1000_C3162 dataset, EFP algorithm able to shrink 14.61% number of items and IST-EFP algorithm able to shrink 26.74%. Found that IST-EFP algorithm is more effective than EFP algorithm in terms of shrinking the number of items, the more the number of items then the IST-EFP algorithm will be more effective on shrinking.

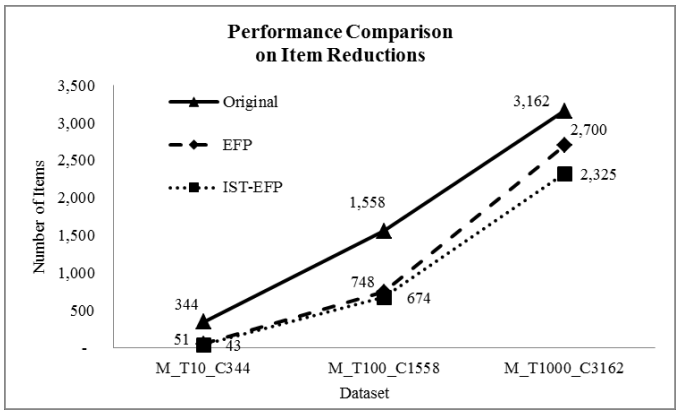

Figure 2

In M_T10_C344 dataset, IST-EFP algorithm execution time about 0.3 seconds faster than the EFP algorithm. In M_T100_C1558 dataset, algorithm execution time IST-EFP about 9 seconds faster than the algorithm EFP. In M_T1000_C3162 dataset, the algorithm execution time is faster IST-EFP about 331 seconds or 5.5 minutes of EFP algorithm. From Figure 3 shows the greater the number of records the execution time difference between the EFP and IST-EFP will be greater where IST-EFP is faster.

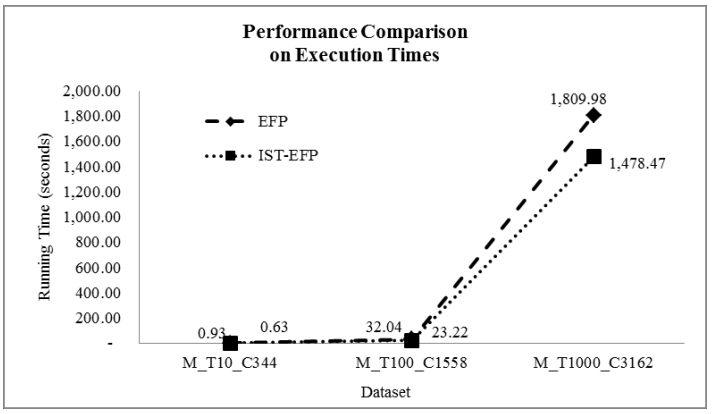

Figure 3

In M_T10_C344 dataset, the number of rules generated by IST-EFP algorithm is 19.71% less than the number of rules generated by EFP algorithm. In M_T100_C1558 dataset, the number of rules generated by IST-EFP algorithm is 3.82% less than the number of rules generated by EFP algorithm. In M_T1000_C3162 dataset, the number of rules generated by IST-EFP algorithm is 1.14% less than the number of rules generated by EFP algorithm. From Figure 4 it appears that the number of rules generated by the EFP algorithm and IST-EFP algorithm is almost the same which means that the information content of rules generated is maintained.

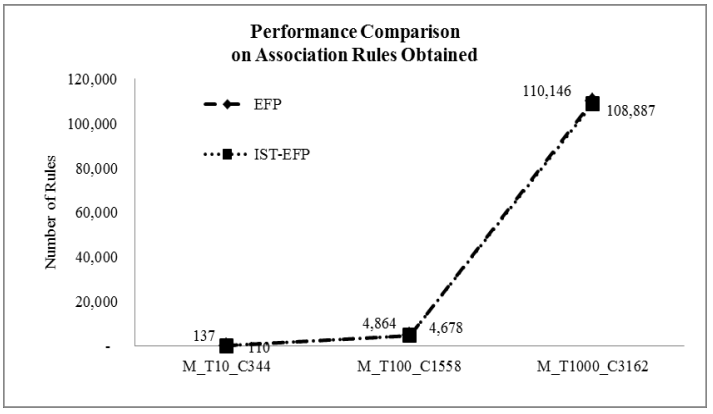

Figure 4

CONCLUSIONS

Of the three datasets were tested, on average IST-EFP algorithm can perform item reduction 13% more, 15% of processing time faster and produces 8% association rules less than the EFP algorithm with information content of rules is maintained.

REFERENCES

[1] B Nath, D K Bhattacharyya and A Ghosh, "Dimensionality Reduction for Association Rule Mining," *International Journal of Intelligent Information Processing,* vol. 2, no. 1, 2011.

[2] T. A. Kumbhare and Prof. Santosh V. Chobe, "An Overview of Association Rule Mining Algorithms," *International Journal of Computer Science and Information Technologies,* vol. 5, no. 1, pp. 927-930, 2014.

[3] A. Sethi and P. Mahajan, "Association Rule Mining: A Review," *The International Journal of Computer Science & Applications,* vol. 1, no. 9, pp. 72-83, 2012.

[4] C. Borgelt, "An Implementation of the FP-growth Algorithm," in *OSDB'05*, Chicago, Illinois, USA, 2005.

[5] L. Vu and G. Alaghband, "Mining Frequent Patterns Based on Data Characteristics," 2012.

[6] M. Suman, T. Anuradha, K. Gowtham and A. Ramakrishna, "A Frequent Pattern Mining Algorithm Based On FP-Tree Structure And Apriori Algorithm," *Research Journal of Computer Systems Engineering,* vol. 02, no. 05, pp. 275-277, 2011.

[7] Y. Sharma and Dr. R.C. Jain, "Analysis and Implementation of FP & Q-FP tree with minimum CPU Utilization in Association Rule Mining," *International Journal of Computing, Communications and Networking,* vol. 1, no. 1, pp. 39-44, 2012.

[8] D. Garg and H. Sharma, "Comparative Analysis of Various Approaches Used in Frequent Pattern Mining," *International Journal of Advanced Computer Science and Applications,* pp. 141-147, 2011.

[9] G. Melli, "The datgen Dataset Generator," [Online]. Available: http://www.datasetgenerator.com . [Accessed July 2014].

.