IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 4, April 2017

Optimal Implementation of CHUD based on Association Matrix

Mr. Lalit N. Dhande[1], Prof. Dinesh D. Patil[2]

M.E. Student, Shri Sant Gadge Baba College of Engineering & Technology, Bhusawal, Maharashtra[1]

HOD (M.E.CSE), Shri Sant Gadge Baba College of Engineering & Technology, Bhusawal, Maharashtra[2]

Abstract: An implementation always has different set of issues, though we have an algorithm or any sort of blueprint ready to solve the problem. This paper describes how CHUD [1] is implemented in C#.Net. Overall four to five main modules we have to form, namely, Connection Module, CHUD Module, DAHU [1] Module, Dataset Population Module and Result Analysis module. Each of the modules has its own significance and contribution. The Connection Module allows us to connect our software to our database file CHUD.accdb as well as to the dataset on which utility mining should be performed. This module will collect the information about columns like ItemSold (which actually contains the itemnames/itemids sold in transaction), transaction Id, item quantity, profit unit etc. As we know CHUD works on promising items only, we need to extract promising items from the given dataset. At the same time while we scan whole dataset we build an association matrix/table which is a vertical representation of given dataset. CHUD module discovers closed high utility itemsets, and records these itemsets in PhaseIOutput table in CHUD.mdb. DAHU later on uses each itemset from PhaseIOutput table and work from the top. Work from the top suggest that it operates on the itemsets of maximum length 'k' and discovers all possible high utility itemsets of length 'k-1', which not present in PhaseIOutput table. All high utility itemsets discovered by DAHU recorded in PhaseIIOutput. The both the tables PhaseIOutput and PhaseIIOutput combinely form a final result set which have all the high utility itemsets for the given dataset. C= {X1, X2……...Xn} where Xi ∈PhaseIOutput. D= {Y1, Y2……. Ym} where Yi ∈PhaseIIOutput. Suppose R is the final result set, then, R=C ∪ D Finally, we represent the result with result pattern module, which use the data of both the table PhaseIOutput and PhaseIIOutput, and represent the items and itemsets with their utility value in Bar chart, PI Chart or any other representation style element. One can make a good analysis with these results.

Keywords: closed high utility itemsets, utility Mining, Concise and lossless representation, data mining.

INTRODUCTION

Data mining is not only the process of extraction of knowledge but also representation of results towards the user. [2-11]Implementation of an algorithm which mines high utility itemsets from the given dataset must be focused on minimum memory usage as well as the time required to execute the main task. The CHUD algorithm works in bottom up manner. Where each individual item gets assessed on the basis of it's utility, and then sort them in a specific order (ascending order of support). The sorted items then considered as nodes. Each of the node then consist of itemset name, PREVSET, POSTSET, tidSet, EstU. Here itemset word is used for set if items which contains single or multiple items. An itemset name is string array which contains the set of item names. PREVSET is an array which contains all itemsets appearing before the current node. POSTSET contains all itemsets appearing after current node. Hence the first node in the list will have PREVSET as null and last node in the list has POSTSET as null. The tidSet is another array which holds the transaction ID's in which itemset is appearing. EstU is a simple variable which hold estimated utility for the current node. All these data members form one node. A node is created for each item on an ordered list.
An AssociationMatrix table is used to represent the original dataset in vertical manner. In which first column is TransID and later columns are the total items in the business. Attribute set for AssociationMatrix table can be shown as A= {TransID,Item1, Item2, Item3,……, Itemn}.
For example;
A= {TransID,'Tea','Cigarrete', 'Chewnngum'}.

A value set can be explained as Av={T1,2,5,2} where 2, 5, and 2 represents the quantity of an item in transaction ID T1. A transaction utility table is used to record the transaction utilities, which have two fields TransID and TUtility, transaction id and transaction utility respectively. Another table UtilityEstimation records itemset utility, which has two fields itemset ID/Name and EstUtility, which is Itemset name and estimated utility of an itemset. We maintain the local transaction utility in LTU table and Global Transaction utility in GTU table.
GTU table holds the utilities of all the transaction, while LTU holds the utilities of the transactions in which current node itemset is appearing.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 4, April 2017

A table named PromisingItems records the promising items in the first scan. These items recorded in PromisingItems table will be used to form an ordered list and then nodes will be created for them. These nodes will send for processing to CHUD algorithm one by one. CHUD algorithm's advantage is that it holds only one node in memory at a time.

Once CHUD completed execution it records closed high utility itemsets in the PhaseIOutput table along with its utility and set length (which helps in PhaseII). We have used comma as delimiter in listing the items in the same set.

TransID	Item_Sold	....	...	...	...
TID1	*I1(3),I2(2),I3(1)*	....	...	...	...
TID2	*I1(2),I2(1),I5(4),I4(3)*	....	...	...	...
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.

OR

TransID	Item_Sold	Quantity	...	...	...
TID1	*I1*	*3*	...	...	...
TID1	*I2*	*2*	...	...	...
TID1	*I3*	*1*	...	...	...
TID2	*I1*	*2*	...	...	...
TID2	*I2*	*1*	...	...	...
TID2	*I5*	*4*	...	...	...
TID2	*I4*	*3*	...	...	...

Fig.1 Transactions in table (Horizontal)

Vertical representation of above dataset can viewed as

TransID	I1	I2	I3	I4	I5
TID1	3	2	1	0	0
TID2	2	1	0	3	4
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.

Fig.2. Transactions in table (Vertical)

Here Fig. 2 shows the vertical representation of dataset present in Fig.1.
This vertical representation is stored in Association Matrix table

Steps to Test the Tool:
To Test the tool, we have developed, we had created a dataset which holds 465 records which are formed on the basis of dataset given as follows

TID	Transaction	Transaction Utility (TU)
T_1	A(1), B(1), E(1), W(1)	5
T_2	A(1), B(1), E(3)	8
T_3	A(1), B(1), F(2)	8
T_4	E(2), G(1)	5
T_5	A(1), B(1), F(3)	11

Fig.3. Trainee Dataset

These five transactions shown in Fig.3. are used to populate the dataset for our tool. Firstly, we convert these five records in vertical manner as shown in Fig. 4. We named A as aitem, B as bitem, E as eitem, F as fitem, W as witem and G as gitem.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 4, April 2017

Sales		
TransId	ItemSold	Qty
1	aitem	1
1	bitem	1
1	eitem	1
1	witem	1
2	aitem	1
2	bitem	1
2	eitem	3
3	aitem	1
3	bitem	1
3	fitem	2
4	eitem	2
4	gitem	1
5	aitem	1
5	bitem	1
5	fitem	3

Fig.4 Vertical Dataset

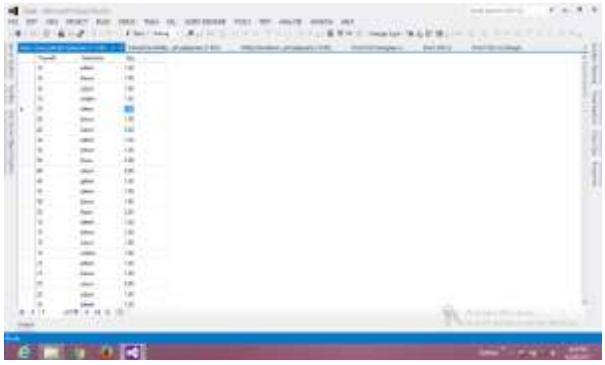

Fig.5. The dataset populated contains 465 records

We insert the same five records in the sales table with different TransactionIDs. The dataset of five transactions get converted to vertical view, it actually spans 15 records. The first module of our tool is populating dataset bases on above 15 records. We enter same 31 data segments with different transaction IDs. Hence the minimum threshold will be now 310 instead 10. The database records are shown in the Fig.5 The records in the table shown in Fig.5

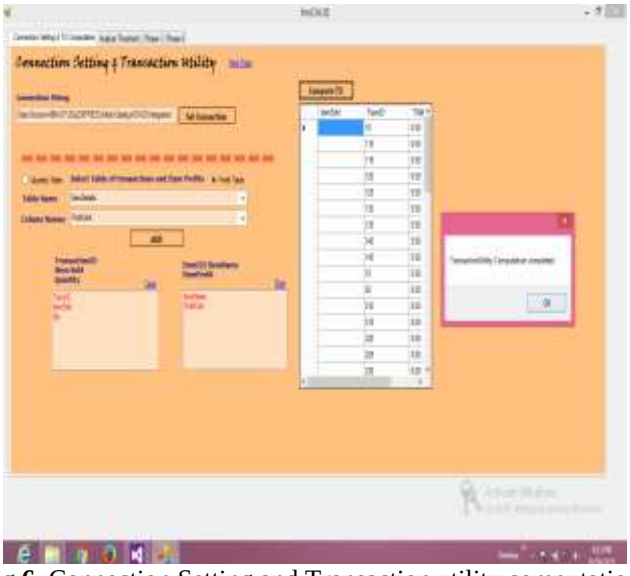

Fig.6. Connection Setting and Transaction utility computation

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 4, April 2017

The first step is to get connected to dataset and get the columns name information required for application of CHUD. Fig. 5 shows the step-1 "Transaction utility computation". We browse the dataset file and set the connection. Clear my file clears the connection of file. Once connection set all the names of tables in database are listed in Table Name Combo Box. The columns of the table are listed in Column Names as we select the table name automatically. Select and add columns TransactionID, ItemSold, and Quantity in left side and same in the right table add ItemName, ProfitUnit columns in right listbox respectively. The button Transaction Utility Computation then pressed for computing the transaction utility of each transaction. Set Column Names link button is used to remember the column names of source dataset file. Then tab Phase-I should be clicked to move on.

This phase takes the input minimum utility and set of items and classifies the items as promising and unpromising labels. The promising and unpromising items are listed in the list boxes separately as shown in Fig.6. Then Phase-I operates further only on promising items. Apply CHUD button to apply CHUD algorithm on promising items. The node information will be displayed. The Phase-I output is hold in Phase-I output list box. After this the output of Phase-I is carried forward to Phase-II in which the closed high utility itemsets are broken into subsets and checked for the high utility condition.

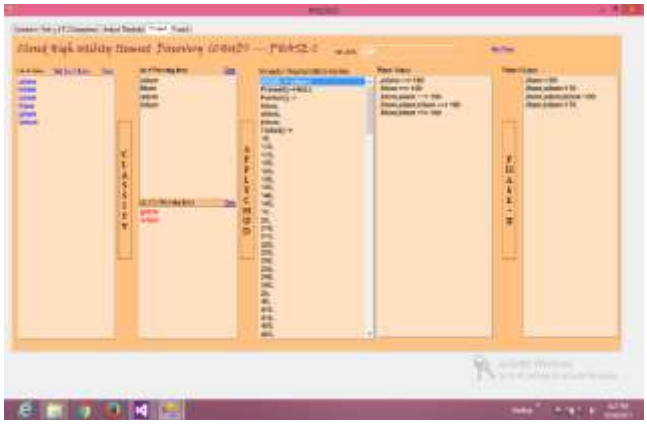

Fig.7. Phase-I (Closed high utility itemset discovery)

In this stage all itemsets are discovered and if any of it satisfies the minimum threshold and not present in the list of Phase-I output, will be added to Phase-II output. The list box named Phase-II output will contain the itemsets of Phase-I and Phase-II both. Then DAHU can be applied with the help of Apply DAHU button. DAHU algorithm takes itemsets from Phase-I and checks for their subsets are high utility itemsets or not. Finally, we get all high utility itemsests listed in Phase-II combobox as shown in Fig.8.

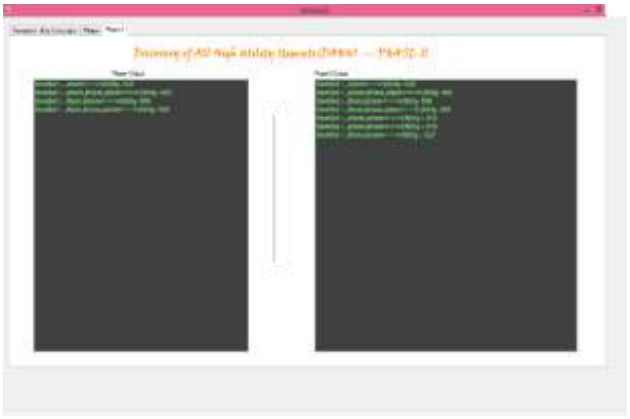

Fig.8. Phase-II (Discovery of All High utility itemset)

CONCLUSION

CHUD algorithm discovers all closed high utility itemsets efficiently. Here 'efficiently' word is used in concerned with compact and lossless representations. Such efficiency of CHUD comes from the very good strategies works with it. This implementation tested only for limited number records and items, but we can conclude that applying CHUD

practically is beneficial. Because it holds only current node in memory, though the node acquires large memory than a simple node.

ACKNOWLEDGMENT

Department of Computer Science & Engineering, Shri Sant Gadge Baba College of Engineering & Technology, Bhusawal, Maharashtra, helped to complete this work.

REFERENCES

[1] Vincent S. Tseng, Cheng-Wei Wu, Philippe Fournier-Viger, and Philip S. Yu, "Efficient Algorithms for Mining the Concise and Lossless Representation of High Utility Itemsets", IEEE
TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 27, NO. 3, MARCH
2015, pp.726-739.

[2] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules", in Proc. 20th Int. Conf. Very Large Data Bases, 1994, pp. 487–499.

[3] C. F. Ahmed, S. K. Tanbeer, B.-S. Jeong, and Y.-K. Lee, "Efficient tree structures for high utility pattern mining in incremental databases", IEEE Trans. Knowl. Data Eng., vol. 21, no. 12, pp. 1708– 1721, Dec. 2009.

[4] J.-F. Boulicaut, A. Bykowski, and C. Rigotti, "Freesets: A condensed representation of Boolean data for the approximation of frequency queries," Data Mining Knowl. Discovery, vol. 7, no. 1, pp. 5–22, 2003.

[5] T. Calders and B. Goethals, "Mining all nonderivable frequent itemsets," in Proc. Int. Conf. Eur. Conf. Principles Data Mining Knowl. Discovery, 2002, pp. 74–85.

[6] K. Chuang, J. Huang, and M. Chen, "Mining top-k frequent patterns in the presence of the memory constraint," VLDB J., vol. 17, pp. 1321–1344, 2008.

[7] R. Chan, Q. Yang, and Y. Shen, "Mining high utility itemsets," in Proc. IEEE Int. Conf. Data Min., 2003, pp. 19–26.

[8] A. Erwin, R. P. Gopalan, and N. R. Achuthan, "Efficient mining of high utility itemsets from large datasets," in Proc. Int. Conf. Pacific-Asia Conf. Knowl. Discovery Data Mining, 2008, pp. 554–561.
738 IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 27, NO. 3, MARCH 2015

[9] K. Gouda and M. J. Zaki, "Efficiently mining maximal frequent itemsets," in Proc. IEEE Int. Conf. Data Mining, 2001, pp. 163–170.

[10] T. Hamrouni, "Key roles of closed sets and minimal generators in concise representations of frequent patterns," Intell. Data Anal., vol. 16, no. 4, pp. 581– 631, 2012.

[11] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2000, pp. 1–12.

BIOGRAPHIES

Lalit N Dhande Student of M. E. (CSE). Interested in data mining related topics, like to implement various algorithms. Classic APRIORI based on association matrix had implemented in C#.Net.

Dinesh D. Patil Working as a HOD-CSE at SSGBCOET Bhusawal, Maharashtra. Interested in research topics related to data mining and knowledge engineering.