Association Rules Extraction from Incremental Databases through ICPT

K. Swarupa Rani
Dept of Computer and Information Sciences
University of Hyderabad
Hyderabad, India
swarupacs@uohyd.ernet.in

V. Kamakshi Prasad
School of Information Technology
Jawaharlal Nehru Technology University Hyderabad
Hyderabad, India
kamakshiprasad@yahoo.com

C. Raghavendra Rao
Dept of Computer and Information Sciences
University of Hyderabad
Hyderabad, India
crrcs@uohyd.ernet.in

Abstract—Association Rule Mining is an important task in data mining. This paper proposes two stage ICPT (Incremental Compact Pattern Tree) Construction Methodology. This methodology facilitates to obtain incidental association rules as well as incremental association rules. This philosophy has been demonstrated with Mushroom Data Set and Pima Indian Diabetes Data Set.

Keywords-Compact Pattern Tree; Association Rule Mining; Incremental Mining; Transactional Data

I. INTRODUCTION

A significant area of data mining research is association rule mining[1] and first introduced by R.Agrawal[2]. An association rule is an implication of the form X→Y, where X and Y are sets of items and $X \cap Y = \emptyset$. The support of this rule is defined as the percentage of transactions that contain the set X, while its confidence is the percentage of these "X" transaction that also contain Y. Extracting association rules from the transactional databases, relational databases and other information repositories which satisfies minimum support and minimum confidence constraints[3] is association rule discovery task. The process of discovering association rules is mainly by finding frequent patterns whose support is greater than the pre-specified minimum support then generating the association rules which satisfy a pre-specified confidence. Finding association rules is valuable for many applications such as cross marketing, attached mailing, catalog design, add-on sales, store layout etc., [3][4].

Incremental mining plays a vital role due to the increasing use of the record-based databases whose data are being continuously added and benefited from previously discovered association rules and carrying further in the process of producing the new set of rules after updates to the database. Several works on incremental maintenance of association rules [5][6][7] are developed which requires the candidate itemsets generation. Han et al.[8] proposes mining frequent patterns without candidate generation, but it is not adaptable to incremental database.

The work presented in this paper is a novel incremental algorithm ICPT(Incremental Compact Pattern Tree). This method facilitates to build for incidental as well as incremental knowledge.

The remainder of the paper is organized as follows. Section II gives a brief review of tree based algorithms. Section III gives the description of proposed algorithms. Section IV gives experimental results. Section V concludes the study.

II. RELATED WORKS

In this section , some incremental updating strategies and tree based algorithms are briefly reviewed. In [5] a data structure, CATS Tree is developed. CATS Tree mines frequent patterns without candidate generation. This is the extention of the work by Han et al. [8] to improve storage compression which turn out to be computationally intensive. Leung et al. proposed CanTree[9] (Canonical Order Tree). The construction of the CANTree only requires one scan. In CANTree items are arranged according to some canonical order, which can be determined by the user prior to the mining process. It depends on two properties. Firstly, the ordering of items is unaffected by the changes in frequency caused by incremental updates. Secondly, the frequency of a node in the CANTree is atleast as high as the sum of frequencies of its children. Although CANTree offers a simple single pass construction process, it yields poor compaction in tree size and spend a large amount of memory space to store the large CANTree.

In[10] INUP-Tree(INcremental Updating Pattern Tree) structure has developed to address updates of the database by scanning only the newly appended transactions, but recomputes all the frequent items from the updated tree, without using the previous knowledge. In[11] Tanbeer et al. proposed iterative reconstruction process for CPTree by considering blocks of data from the given database. It mainly consists of two phases. Firstly, insertion phase, that inserts transaction into tree according to items appearance order. Secondly restructuring phase, restructures according to frequency descending order of items. This process is continued till end of the transaction in the database. Due to repeated process it consumes more time to construct the tree for whole database. Later, Vadivel et al. [12] proposed modified Compact Pattern Tree by considering whole database as one block which incidentally address the time complexity. In this paper, proposes a novel tree structure ICPT(Incremental Compact Pattern Tree) for incremental association rule mining is the extension of the work[13]. ICPT builds incidental knowledge along with updated incremental knowledge. The proposed method and the algorithms are discussed in further sections.

III. INCREMENTAL COMPACT PATTERN TREE

In this paper we proposed ICPT Methodology consisting of two phases (i) Organizing and (ii) Merging. ICPT provides in order to mine association rules for incremental database by obtaining frequent pattern from ICPT through updated item sort list. Table I shows the notation description used in algorithms and illustrations. Table II shows the sample transactions.

TABLE I

Notations Description

Name	Description
TT	Transactional Tree
RT	Restructure Tree
ICPT	Incremental Compact Pattern Tree
ECPT	Existing Compact Pattern Tree
CCPT	Current Compact Pattern Tree
ES	Existing item sorted list
CS	Current item sorted list
O_ECPT	Organizing Existing Compact Pattern Tree
O_CCPT	Organizing Current Compact Pattern Tree
US	Updated item sorted list
EL	Existing item appearance order list(unsorted list)
CL	Current item appearance order list(unsorted list)
CPT	Compact Pattern Tree
IS	Item sorted list
O_CPT	Organizing Compact Pattern Tree
Branch b_i	Branch of the tree from root to i^{th} leaf node
TID	Transaction unique identifier
‘items’	Items were purchased during a particular transaction

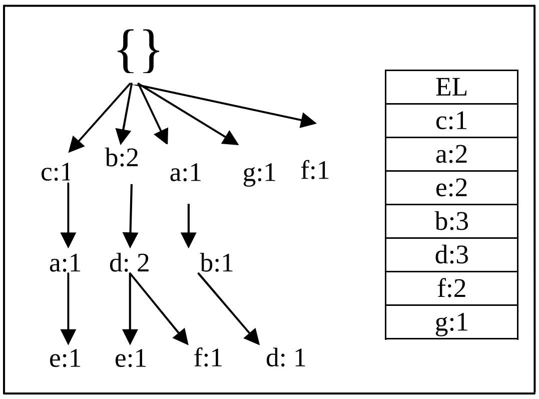
ICPT computation requires two phases. The first phase involves Organizing the ECPT and CCPT through US (union of ES and CS). The second phase consists of Merging O_ECPT and O_CCPT.

The approach starts with constructing TT by maintaining EL. From Table II assuming the transaction TID's from 10 to 60 are existing transactional database and the remaining TID's from 70 to 90 are appended transactions such as current transactional database. Constructing TT from existing transactional database by maintaining EL as shown in Fig 1. EL sorted in descending order and maintained as ES. Based on ES the RT of the tree takes place and shown in Fig 2.(ECPT).

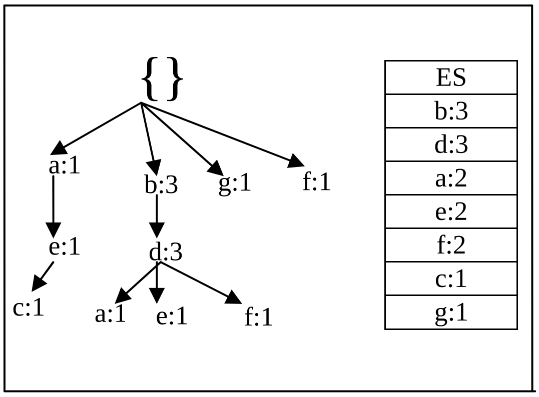
TABLE II

Sample Transactional database

TID	items
10	c,a,e
20	b,d,f
30	a,b,d
40	b,d,e
50	f
60	g
70	f,e
80	c,f,e
90	f,e,d



EL
c:1
a:2
e:2
b:3
d:3
f:2
g:1

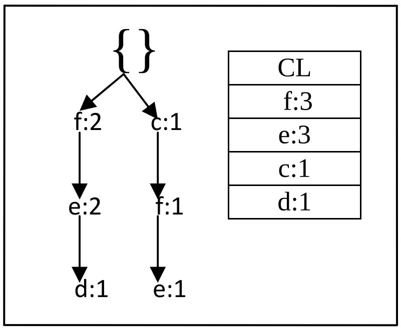


ES
b:3
d:3
a:2
e:2
f:2
c:1
g:1

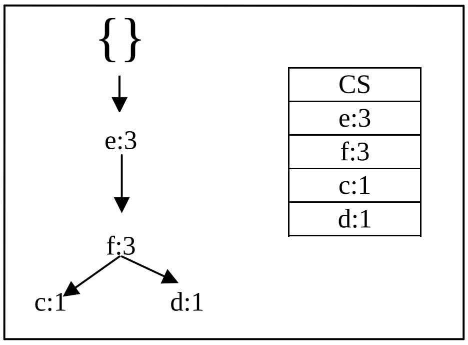
Figure 1. TT of TIDs 10 to 60

Figure 2. ECPT-RT according to ES of Fig 1.

In the following, we describe how to update the tree structure and find all the frequent itemsets when a set of transactions is added. Treating the set of appended transactions as a stand alone current database, the procedure described above is applied to construct CCPT. CCPT is used for deriving incidental knowledge. To illustrate for appended transactions, constructing TT from current transactional database by maintaining CL as shown in Fig 3. then restructured according to CS shown in Fig 4(CCPT). Therefore, by applying FP_Growth Mining Technique[8] to CCPT derives incidental knowledge. In order to obtain incremental knowledge, ICPT is to be constructed by applying proposed algorithms given in the following Sections with appended transactions.



CL
f:3
e:3
c:1
d:1



CS
e:3
f:3
c:1
d:1

Figure 3. TT of TIDs 70 to 80

Figure 4.CCPT-RT according to CS of Fig 3.

The following are the description of frequent patterns and association rules of CCPT

A. Frequent Patterns

A Pattern(a set of items, subsequences, substructures, etc.) that occurs frequently in a data set

Frequent Patterns generated with min.sup threshold 30% from CCPT are:
{d,c,f,e,(d f),(d e),(c f),(c e),(f e),(d f e),(c f e)}

B. Association Rules

Association rules are statements of the form{X_1,X_2,….X_n}→ Y, meaning that if we find all of X_1,X_2,….X_n in the market basket, then we have a good chance of finding Y. The probability of finding Y in order to accept the rule is called the confidence of the rule. Searching only for rules that had confidence above a certain threshold.[14].

Association Rules extracted with min. conf. threshold 45% from CCPT(incidental knowledge) are:
{d→f,d→e,c→f,c→e,e→f,f→e}

As discussed earlier ICPT requires two phases (i) Organizing phase and (ii) Merging phase. By combining ES and CS produces the list which is to be sorted in frequency descending order and maintained as US as shown in Fig 6.

The first phase, Organizing is described. ECPT and CCPT has to be restructured based on US. The “Algorithm for Organizing Compact Pattern Tree” reorders ECPT and CCPT by using US and new compact pattern trees are generated such as O_ECPT and O_CCPT as shown in Fig 5 and Fig 6 respectively(note: there is no change in figure 6 because US item frequency order is same as CS).

The second phase, Merging is described. O_ECPT and O_CCPT merged by applying the proposed algorithm in Section E. In order to merge the above two organizing trees such as O_ECPT and O_CCPT, The “Algorithm for Merging Organized Compact Pattern Trees” merges both the trees by taking each branch in O_CCPT and finding common nodes in O_ECPT relative to the branch of O_CCPT. The algorithm is followed and illustrated in Fig 7. ICPT which is the Incremental Compact Pattern Tree.

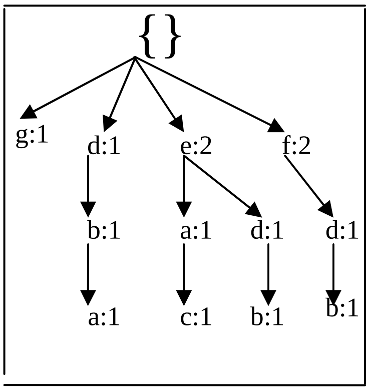
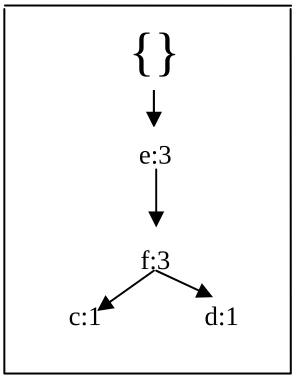


Figure 5. O_ ECPT according to US of Fig 3.



US
e:5
f:5
d:4
b:3
a:2
c:2
g:1

Figure 6. O_CCPT according to US of Fig 4.

Association rules are usually required to satisfy a pre-specified minimum support and a pre-specified minimum confidence at the same time. Therefore, by applying FP_Growth Mining Technique[8] to Fig 7. ICPT the frequent patterns are generated. With minimum confidence constraint to these frequent patterns Incremental Association Rules are generated.

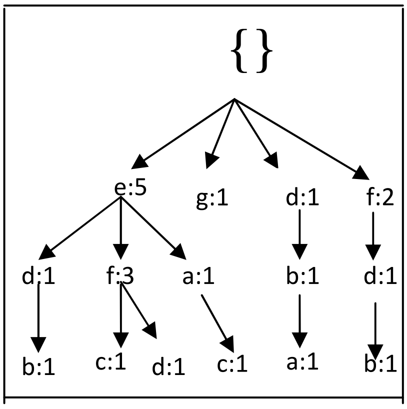


Figure 7. ICPT-Incremental Compact Pattern Tree

Frequent Patterns generated from ICPT with min.sup threshold 30% are:{b,d,f,e,(b d),(f e)}
Association Rules extracted from ICPT(incremental knowledge) with min. conf. threshold 45% are:{d→b,b→d,f→e,e→f}

In the following, Section C gives proposed algorithm for Incremental Compact Pattern Tree. Section D gives an algorithm for Organizing ECPT and CCPT. Section E gives proposed algorithm for Merging O_ECPT and O_CCPT.

C. An algorithm for Incremental Compact Pattern Tree

```
Algorithm: Incremental Compact Pattern Tree
Input: ES, CS,ECPT,CCPT
Output:ICPT
Begin
(1)a)  US=Combine ES and CS
   b)  sort the US in descending order
(2)   O_ECPT= OrganizeCPTree(ECPT,US)
(3)   O_CCPT= OrganizeCPTree(CCPT,US)
(4)    ICPT=MergeCPTree(O_ECPT,O_CCPT)
End
```

D. An algorithm for Organizing Compact Pattern Tree

```
Algorithm:Organizing Compact Pattern Tree
OrganizeCPTree
Input: CPT,IS
Output:O_CPT
Begin
(1)   for each branch $b_i$ in CPT
(2)   remove the branch $b_i$ from CPT
(3)   sort the nodes according to IS
(4)   insert sorted $b_i$ into O_CPT
End
```

E. An algorithm forMerging Organized Compact Pattern Trees

```
Algorithm:Merging Organized Master and Organized Current Compact Pattern Tree
MergeCPTree
Input: O_ECPT,O_CCPT
Output:ICPT
Begin
(1) Initialize ICPT = O_ECPT
(2) for each branch $b_i$ in O_CCPT
(2) find branch $b_j$ in ICPT having common nodes at
     beginning with $b_i$
```

```
(3) If $b_j$ exists then
      Let $b_i = s.y_i$ , $b_j=s.y_j$ where ‘s’ is maximal prefix containing
      the common beginning nodes of $b_i,b_j$
(4)  If ‘s’ is nonempty
      Update frequencies of ‘s’ in $b_j$ path with the frequencies of
      ‘s’ in $b_i$ path of O_CCPT
     else
     Add $b_i$ as a new branch to the root node of ICPT with
its frequencies
End
```

IV. EXPERIMENTAL RESULTS

To evaluate the performance of proposed approach, conducted experiments on benchmark datasets such as mushroom and pima Indian diabetes dataset. The implementation of the existing and proposed algorithms is done using Java Programming Language. The PC Configuration includes CPU Intel (R) Pentium(R) Dual T2330 1.60 GHz, 1GB RAM in Windows 7. In the mushroom dataset (with 1000 transactions) considering the first 700 transactions as existing database and remaining 300 as current transaction database where as in Pima Indian Diabetes data set (with 500 transactions) considering the first 300 as existing and last 200 as current. The data is converted into transactional database suitable to the experiments. The following tables describes the experimental results for incremental and incidental knowledge such as ICPT and ECPT,CCPT and also describes with various support threshold values of ICPT.

TABLE III
Experimental Results for MushRoom Data Set for various min.support and min.confidence thresholds for ICPT

Support	Confidence	Time(s) for extracting Frequent Patterns	Time(s) for extracting Association Rules	Total time(s)	No. of Association Rules extracted
.40	.60	.022	12.118	12.236	136973
.45	.65	.017	6.013	6.121	86270
.50	.70	.015	3.011	3.116	49128
.55	.75	.013	1.314	1.416	34333
.60	.80	.011	.839	.933	21627

TABLE IV
Experimental Results for Pima Indian Diabetes Data Set for various min.support and min.confidence thresholds for ICPT

Support	Confidence	Time(s) for extracting Frequent Patterns	Time(s) for extracting Association Rules	Total time(s)	No. of Association Rules extracted
.40	.60	.035	9.416	9.478	134088
.45	.65	.025	4.210	4.260	93931
.50	.70	.023	3.491	3.537	59240
.55	.75	.019	1.134	1.174	36974
.60	.80	.015	.511	.442	27721

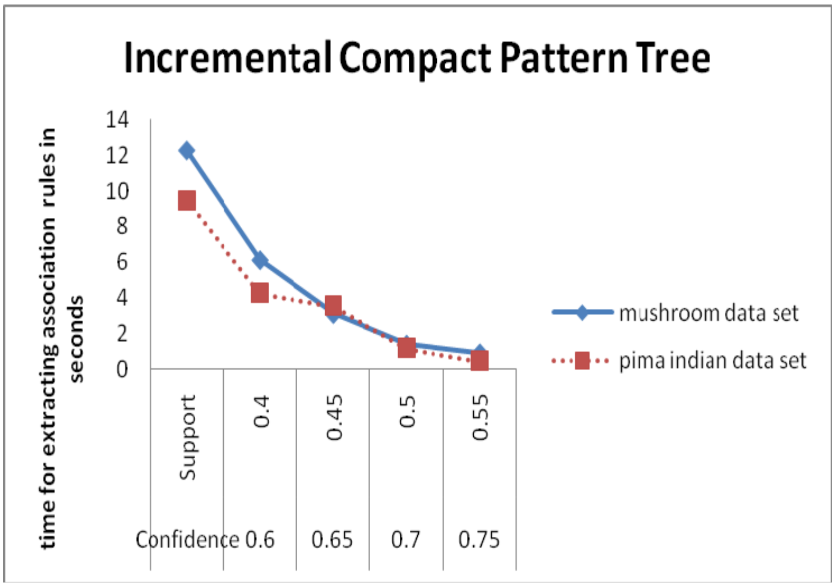


Figure 8. Execution Time for extracting Association Rules from ICPT

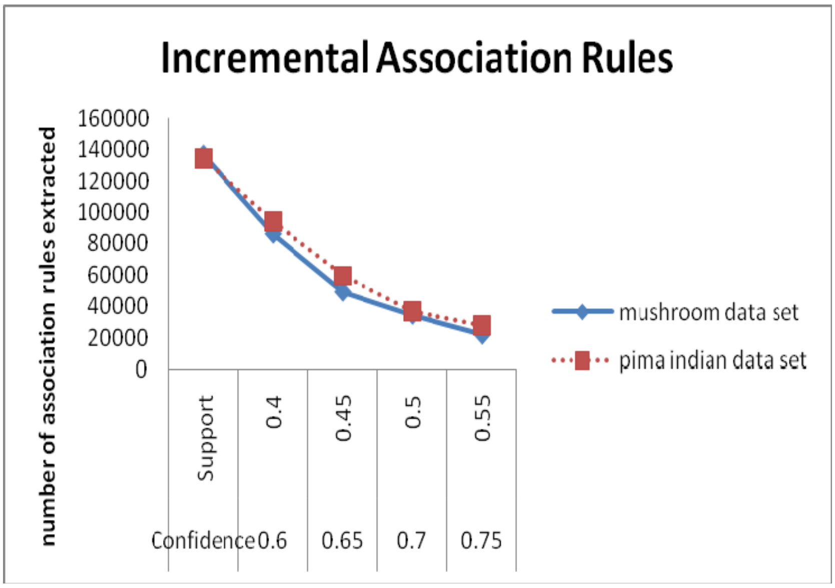


Figure 9. Number of Association Rules extracted from ICPT

TABLE V

Experimental Results for Mushroom Dataset for building incidental and incremental knowledge with min.support 40% and min.confidence 60%

	Time(s) for TT	Time(s) for RT	Time(s) for OT	Time(s) for merging OT's of ECPT & CCPT	Time(s) for extracting Frequent Patterns	Time(s) for extracting Association Rules	Total time(s)	No. of Association Rules extracted
ICPT	**-**	**-**	**-**	**.004**	**.024**	**11.992**	**12.02**	**35388**
ECPT	**.016**	**.031**	**.016**	**-**	**.018**	**9.960**	**10.025**	**40964**
CCPT	**.006**	**.010**	**.008**	**-**	**.014**	**6.893**	**6.923**	**25566**

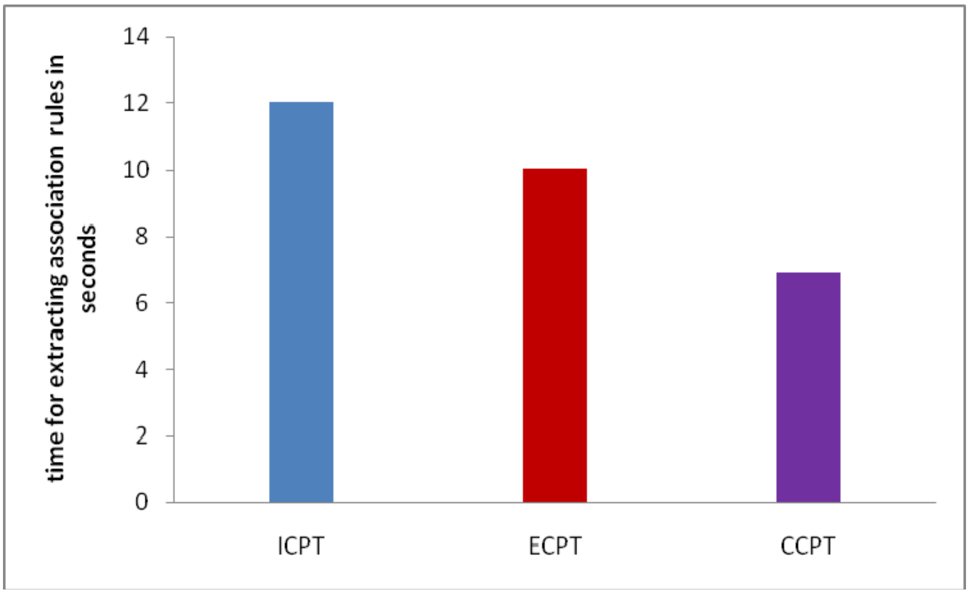


Figure 10. Execution Time for extracting Association Rules from ICPT,ECPT,CCPT using Mushroom Data Set

TABLE VI

Experimental Results for Pima Indian Diabetes Dataset for building incidental and incremental knowledge with min.support 40% and min.confidence 60%

	Time(s) for TT	Time(s) for RT	Time(s) for OT	Time(s) for merging OT's of ECPT & CCPT	Time(s) for extracting Frequent Patterns	Time(s) for extracting Association Rules	Total time(s)	No. of Association Rules extracted
ICPT	-	-	-	.003	.015	8.193	8.211	38175
ECPT	.014	.018	.012	-	.013	6.251	6.296	40964
CCPT	.012	.014	.008	-	.009	4.964	4.999	42844

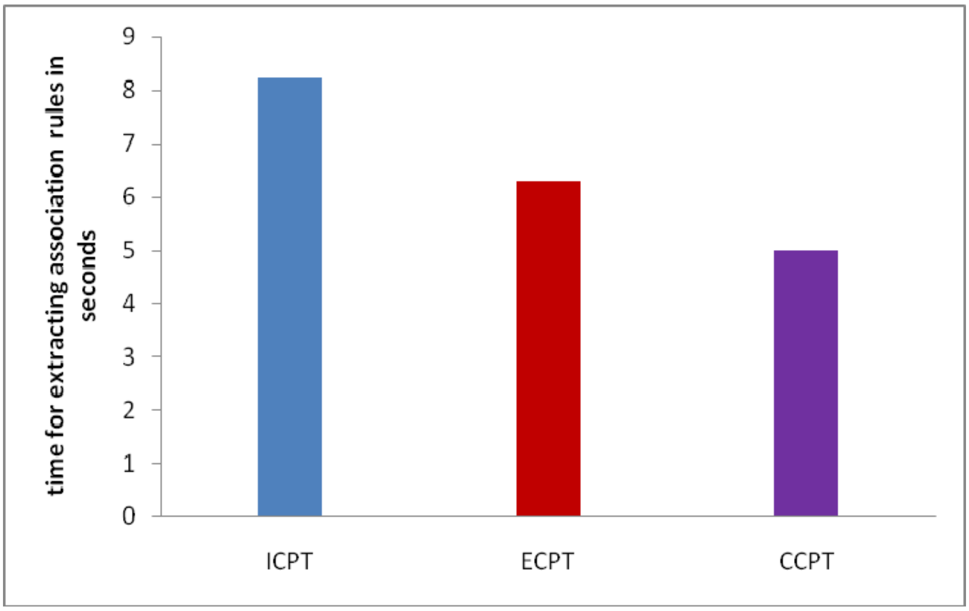


Figure 11. Execution Time for extraction Association Rules from ICPT,ECPT,CCPT using Pima Indian Diabetes Data Set

V. CONCLUSIONS

The proposed methodology is to extract association rules based on various tree based algorithms by extracting frequent patterns in incremental mining. The incremental transactions knowledge can be derived independently which minimizes intercoupling issues. This will have extreme benefits when incremental transactions(logs) are huge as well as when distributed. One can plan scalable and/or real time frequent patterns extraction system. The work in this direction is in progress.

ACKNOWLEDGMENT

We thank Dr.Rajeev Wankar and P.S.V.S Sai Prasad of University of Hyderabad, for constructive discussions.

REFERENCES

[1] Mafruz Zaman Ashrafi, David Taniar and Kate Smith, “ODAM: An Optimized Distributed Association Rule Mining Algorithm”, IEEE Distributed Systems Online 1541-4922,2004, Published by IEEE Computer Society Vol. 5, No.3, March 2004, pp.7-9,15.

[2] R. Agrawal T.Imielinkski and A.Swami: Mining Association Rule Between sets of items in large database, Proceedings of ACM SIGMOD, page 207-216, May 1993.

[3] Agrawal, R. and Srikant, R.”Fast algorithms for mining association rules”.In:Proceedings of the 1994 International Conference On Very Large Data Bases (VLDB’94), Santiago, Chile, pp 487-499, 1994.

[4] Agrawal, R.; Imielinskil, T.; and Swami, A. Mining Associations between Sets of Items in Massive Databases. Proc. Of the ACM SIGMOD Int’l Conference on Management of Data, pp.207-216, Washington D.C., May 1993.

[5] Cheung, D.W., Han, J., Ng, V.T., Wong, C.Y.: Maintenance of Discovered Association Rules in Large Database: An Incremental Updating Technique, In Proceedings of the 12th International Conference on Data engineering, New Orleans, Louisiana, 1996.

[6] Cheung, D.W., Lee, S.D., Kao, B.: A General Incremental Technique for Maintaining Discovered Association Rules, In Proceedings of the Fifth International Conference on Database Systems for Advanced Applications, Melbourne, Australia, Jan, 1997.

[7] Zhou, Z., Ezeife, C.I.: A Low-Scan Incremental Association Rule Meintenalnce Method Based on the Apriori Property, in the Proceedings of the fourteenth Canadian Conference on Artificial Intelligence, AI 2001, June 2001, Ottawa.

[8] J.Han, J.Pei, and Y.Yin, “Mining Frequent Patterns without Candidate Generation,”Proc. ACM Special Interest Group on Management of Data(ACM SIGMOD 00), June. 2000, pp.1-12, doi:10.1145/342009.335372

[9] Carson Kai-Sang Leung, Quamrul I.Khan and Tariqul Hoque, “CanTree: A Tree Structure for Efficient Incremental Mining of Frequent Patterns,” in Proceedings of the 5th IEEE International Conference on Data Mining(ICDM),pp.274-281,2005.

[10] Hai, T.H., Shi, L.Z.: A New Method for Incremental Updating Frequent Patterns Mining. In: Proceedings of the Second International Conference on Innovative Computing, Informatio and Control, p.561(2007)

[11] Tanbeer,. S.K., Ahmed, C.F.,Jeong,B.S.,Lee,Y.K: Efficient single-pass frequent pattern mining using a prefix tree. Information Sciences 179,559-583(2008)

[12] R.Vishnu Priya,A.Vadivel and R.S.Thakur L.Cao,J,Zhong, and Y.Freng(eds.):ADMA 2010, Part I,LNCS 6440, Springer-Verlag Berlin Heidelberg pp.254-261,2010.

[13] K. Swarupa Rani, V.Kamakshi Prasad, C. Raghavendra Rao “Mining Frequent Pattern through Incremental Compact Pattern Tree” unpublished. Presented orally in International Conf. on Advancements in Engineering and Management, RITS,2012

[14] http://infolab.stanford.edu/~ullman/mining/assocrules.pdf

AUTHORS PROFILE

K. Swarupa Rani obtained her MCA from Sri Krishnadevaray University and M.Phil(Computer Science) from Madhurai Kamaraj University. Currently, she is pursuing her Ph.D in NU. She is now Assistant Professor from Department of Computer and Information Sciences in University of Hyderabad. She has the experience of guiding around 24 post graduate students (M. Tech/M.C.A).Her research interests are in data mining, time series, text processing.

Prof . V. Kamakshi Prasad, born on 02-08-1968, obtained his PhD from IIT Madras in the year 2002, received his M.Tech. in Computer Science from Andhra University and B. Tech. in Civil Engg. from KL College of Engg. Nagarjuna University. Joined in JNT Univeristy Hyderabad, as lectures in the year 1992, promoted as Associate Professor and Professor in the years 2003 and 2006 respectively. Research areas are Speech processing, pattern recognition and Data mining.

Prof. C. Raghavendra Rao, completed his B. Sc and M.Sc in Statistics from Andhra University and Osmania University respectively. Ph. D. in Statistics and M.Tech(CS&Engineering) from Osmania University. He worked as a lecturer in Statistics at Osmania University. Since 1986, he is working in the School of

Mathematics and Computer/Information Sciences, University of Hyderabad. Presently he is a Professor in the Dept. of Computer and Information Sciences, University of Hyderabad.His current research interests are Simulation & Modeling and Knowledge Discovery. Dr Rao is a member of the Operation Research Society of Indian Mathematical Society, International Association of Engineers, Society for development of statistics, Andhra Pradesh Society for Mathematical Sciences, Indian Society for Probability and Statistics, Indian Mathematical Society and Society for High Energy Materials, International Rough Set Society, Indian Society for Rough sets also a Fellow of The Institution of Electronics and Telecommunication Engineers and Society for Sciences. Dr Rao Guided 5 PhDs, 25 M.Techs, 8 M.Phils. Nearly 40 Journal and 80 Proceeding Papers to his credit.Dr. Rao contributed enormously in the project of National Interest, few to mention are:(1) Mathematical model based Control system for swing out operation for the liquid operation feeding boom for rocket launching pad in Sree Hari Kota (T.A. Hydraulics Ltd.),(2)Design of Optimal driving profile generation for diesel locomotives for Indian Railways (Consultancy project for Medha Servo Control System). (3) Mathematical modeling for 6 degree of Motion platform for building driver training simulator.(4) Mathematical modeling fuel air explosive system for Defense Research Development Organization.(5)Optimal design configuration methodologies for aerospace applications with IIT Bombay for DRDL.