IJARSE, Vol. No.4, Special Issue (02), February 2015

ISSN-2319-8354(E)

# EFFICIENT MINING OF HIGH UTILITY ITEMSETS FROM TRANSACTIONAL DATABASES

# Mr. Vivek J. Jethe<sup>1</sup>, Prof. Manoj D. Patil<sup>2</sup>, Prof. Sachin Chavan<sup>3</sup>

<sup>1,2,3</sup>Computer Department, MGMCET, Mumbai university, (India)

# ABSTRACT

In this paper, we would like to show you the working of Hadoop and MapReduce in terms of Data mining. There were many methods proposed earlier for this purpose. Data mining is concerned with analysis of large volumes of data to automatically discover interesting regularities or relationships which in turn leads to better understanding of the underlying processes. The primary goal is to discover hidden patterns, unexpected trends in the data. Data mining activities uses combination of techniques from database technologies, statistics, artificial intelligence and machine learning. Methods like Apriori algorithm, Frequent pattern mining were implemented for this purpose which were not very efficient in terms of time, space and certain parameters like weighted items. These limitations were overcome by bringing into picture the concept of Hadoop and MapReduce.

# Keywords : Hadoop, Mapreduce, Data Mining, Utility of Itemsets, Weight

#### I. INTRODUCTION

To provide the efficient solution to mine the large transactional datasets, recently improved methods are presented in [1]. In [1], authors presented propose two novel algorithms as well as a compact data structure for efficiently discovering high utility itemsets from transactional databases. Experimental results show that UP-Growth and UP-Growth+ outperform other algorithms substantially in terms of execution time. But these algorithms further needs to be extend so that system with less memory will also be able to handle large datasets efficiently. The algorithms presented in [1] are practically implemented with memory 3.5 GB, but if memory size is 2 GB or below, the performance will again degrade in case of time. As when the size of the database increases and if the memory is low, problem arises. We cannot expect the database to be of limited size at least in the real world where commercial aspects are taken into consideration. In the real commercial world, customers are increasing in numbers and their transactions are also increasing and if our database is limited then we are limiting our business which is a shear considerable loss in terms of customer relationship, finance, etc. So to meet such a demand we should be using such a technology that without upgrading the hardware configuration of our systems, we should be able to meet the ever increasing needs of the business growth. Small up gradation would not matter a much for a small business but when it comes to upgrading the whole system, it costs a lot. In this project we are presenting new approach which is extending these algorithms to overcome the limitations using the MapReduce framework on Hadoop.

#### IJARSE, Vol. No.4, Special Issue (02), February 2015

ISSN-2319-8354(E)

#### II. UP-GROWTH ALGORITHM

To generate these high utility itemsets mining recently in 2010, UP - Growth (Utility Pattern Growth) algorithm [2] was proposed by Vincent S. Tseng et al. for discovering high utility itemsets and a tree based data structure called UP - Tree (Utility Pattern tree) which efficiently maintains the information of transaction database related to the utility patterns. Four strategies (DGU, DGN, DLU, and DLN) used for efficient construction of UP - Tree and the processing in UP - Growth [11]. By applying these strategies, can not only efficiently decrease the estimated utilities of the potential high utility itemsets (PHUI) but also effectively reduce the number of candidates. But the problem with this algorithm is that this algorithm takes more execution time for phase II (identify local utility itemsets) and I/O cost.

Efficient discovery of frequent itemsets in large datasets is a crucial task of data mining. Lately, several approaches have been proposed for generating high utility patterns; they lead to the problems of producing a large number of candidate itemsets for high utility itemsets and probably degrade mining performance. Mining high utility itemsets from a transactional database refers to the discovery of itemsets with high utility like profits. Though a number of relevant approaches have been proposed in recent years, they lead to the problem of producing a large number of candidate itemsets for high utility itemsets. If there is such a large number of candidate itemsets, it degrades the mining performance in terms of execution time and space requirement. This situation may become worse when the database contains lots of long transactions or long high utility itemsets.

Existing studies applied overestimated methods to facilitate the performance of utility mining. In such methods, potential high utility itemsets (PHUIs) are found first, and thus an additional database scan is performed for utilities' identification. However, existing methods often generate a huge set of PHUIs and their mining performance is degraded consequently. This would become worse when databases contain many long transactions or low thresholds are set. The large number of PHUIs forms a challenging problem to the mining performance since the more PHUIs the algorithm generates, the processing time consumed is very high..

#### III. OVERVIEW OF HADOOP AND MAP/REDUCE ARCHITECTURE



29 | Page

#### IJARSE, Vol. No.4, Special Issue (02), February 2015

#### ISSN-2319-8354(E)

As shown above, is the architecture of the Hadoop and Map/reduce. There's a heap of data in the first stage as shown in the figure. This data is then processed by the hadoop cluster where we have some mappers and there's a reducer. For each DFS blocks, a mapper is assigned and the result of all the mappers is computed in the reducer. In the reducer stage, the results are computed. So the desired result is achieved.

#### IV. PERFORMANCE COMPARISON

To understand it better, we will consider a particular dataset as shown in the flowchart. After that, we will apply up-growth and up-growth+ and then calculate the time required. Then apply up-growth and up-growth+ with Hadoop and then calculate its time. Compare the results in terms of time.



#### V. THE MAPREDUCE FRAMEWORK FOR HANDLING BIG DATASETS

MapReduce is a programming model for processing large data sets with a parallel, distributed algorithm on a cluster. A MapReduce program consists of a Map() procedure that performs filtering and sorting (such as sorting students by first name into queues, one queue for each and every name) and a Reduce() procedure that performs a summary operation (such as counting the number of students in each queue, which yields name frequencies). The "MapReduce System" (also called "infrastructure" or "framework") orchestrates by marshalling the distributed servers, running the various tasks in parallel, which manages all communications and data transfers between the various parts of the system, which provides redundancy and fault tolerance.

"**Map**" **step**: The master node takes the input, and then divides it into smaller sub-problems, and distributes them to worker nodes. A worker node repeats this step in turn, leading to a multilevel tree structure. The worker node processes the smaller problem, and passes the answer back to its master node.

"**Reduce**" **step:** Master node then collects the answers to all the sub-problems and combines them in some way to form the output – the answer to the problem it was originally trying to solve.

# IJARSE, Vol. No.4, Special Issue (02), February 2015

ISSN-2319-8354(E)

#### VI. HADOOP OVERVIEW

When data sets go beyond a single storage capacity, then distributing them to multiple independent computers becomes important. Trans-computer network storage file management system is called A distributed file system is Trans-computer network storage file management system . A typical Hadoop distributed file system contains thousands of servers, where each server stores partial data of file system.

Hadoop enables a computing solution that is:

- Scalable– New nodes can be added as needed, and added without needing to change data formats, how data is loaded, how jobs are written, or the applications on top.
- Cost effective– Hadoop brings massively parallel computing to commodity servers. The result is a sizeable decrease in the cost per terabyte of storage, which in turn makes it affordable to model all your data.
- Flexible– Hadoop is schema-less, and can absorb any type of data, structured or not, from any number of sources. Data from multiple sources can be joined and aggregated in arbitrary ways enabling deeper analyses than any one system can provide.

Fault tolerant– When you lose a node, the system redirects work to another location of the data and continues processing without missing a beat.

0.97 \$.0.d.	「原意義、 <b>開始後、当</b> 者」前は合いた。	El Map-Rattuce
Project Explore (	<pre>M D &amp; teclars II Mag.Rebujow BacOunDencolies M mainframejou B UgutecLjow *** To channe this template, stoose Tools   Templates pookage treefest; *** : *******************************</pre>	B Outline 1 P C V V V V V V V V V V V V V V V V V V V
	T. Fordern R. Tala . In Junity & Man Refered Landson .	
	No consolar to display at this time.	

#### VII. EXPERIMENTAL SETUP

The above screenshot is of the eclipse screen where we have just started and trying to run the code.

	International J	Journal o	of Advance	Research	In Science A	and Engineer	ring	http://w
--	-----------------	-----------	------------	----------	--------------	--------------	------	----------

IJARSE, Vol. No.4, Special Issue (02), February 2015

ISSN-2319-8354(E)

A REAL PROPERTY AND A REAL		
Min Item	ng High Utility sets from Transactional Data	bases
	Browse Dataset O without Hadoop O With I	Hadbop
Formatted dataset with Transactional Utility Show P	Clear	
Enter Min Support(%)	Transaction weighted utility Softed Transaction	
Transaction weighted utility	Transactions with Utility	

The above screen appears when you run the Map/Reduce on eclipse which consists of the code.On this console, we are going to run the test on the datasets with and without Hadoop with Up-growth and Up-Growth+.

# VIII. TESTING WITHOUT HADOOP

A TOTAL PROPERTY AND ADDRESS OF TAXABLE PARTY.	the second se	
	Mining High Utility Itemsets from Transactional Databases	2
Preprocessing UP-Growth		
UsersWI-kc/DesittopWvek project/transds	ataset110ir1.txt Browse Dataset () With Hadoop	
Formated dataset with Transactional C	Snow Result   Clear	
Enter Min Sunnort(%)	Transaction weighted utility Social Transactions	
Luci min support(s)		
Transaction weighted	New Transactions with Utility	
Transaction weighted	ity Transactions with Culity	
142=487		
114=159	T0->(121,4) (421,7) (134,4) (243,2) (411,9) (321,10) (424,1) (444,2) (332,1) (313,2) (214,4)	(331,4) (333,9) (434,5
323=294	T1->(233,5) (121,7) =22	
334=298	T2->(441.2) (221,2) (313,2) (434,1) (414,1) (344,1) (444,8) (213,1) (134,9) (411,1) (433,2) (	121,5) (413,5) =132
244=163	T3->(134.6) (414.4) (312.5) (324.2) (242.4) (223.1) (422.5) (221.2) (243.3) (233.3) (344.4) (	331,1) (122,3) (142,1
442=103	T4->(344,11) (334,2) (332,3) (411,1) (414,5) (244,5) (321,7) (442,3) (232,3) (233,3) (133,8)	(313,1) (443,1) (131,
122-256	10*9(335,1)(142,2)(451,7)(343,1)(321,4)(444,2)(315,1)(232,5)(334,3)(314,3)(422,1)( 15,5)(44,4)(42,4)(42,1)(42,1)(42,4)(41,4)(41,4)(42,4)(4	332,17 (312,2) (413,4
443=163	10**(21,11)(123,1)(131,4)(142,3)(241,4)(411,1)(412,1)(134,1)(231,3)(143,3)(133,10 17.5(42,21)(43,4)(131,7)=31	(3(4,0)(422,0)(22)
241=356		
431=135		
343=135		
314=328		
423=135		
143=328		
211=193		
412-103		
212=193		
And a state of the		

The first screen is the loading of the dataset on the console. The first attempt of ours would be to perform data mining without Hadoop and just with Up-Growth and Up-Growth+.

The next step would be to find out the transaction weighted utility by pressing on the respective button on the console. After pressing the button, we get the following output on the screen.

IJARSE, Vol. No.4, Special Issue (02), February 2015

# ISSN-2319-8354(E)

Mining Itemset	High-Utility s from Trensactional Databases
Prozenokasing UP-Grewin	
Inserve eccentropress projects an adaptation of the line Formated dataset with transactional Unity   Show Result Enter Min Support(%) 30 Transaction weighted utility	Transactions with Utility
344         = 412           4T4         = 412           203         = 337           204         = 337           204         = 337           203         = 337           204         = 337           203         = 337           204         = 337           203         = 344           232         = 234           232         = 234           234         = 340           443         = 310           443         = 310           443         = 3114           425         = 3114           425         = 3114           425         = 3114           521         = 304           231         = 304           231         = 304           231         = 304           231         = 304           234         = 304           234         = 306           232         = 304           234         = 306	$ \begin{array}{c} T0 = (121,4) (421,7) (134,4) (243,2) (414,0) (221,10) (424,1) (332,1) (313,2) (214,4) (331,4) (333,0) (424,2) \\ T1 = (233,0) (231,7) (322,2) (231,7) (322,2) (231,7) (234,6) (233,7) (234,6) (231,7) (234,6) (231,7) (232,7) (234,6) (231,7) (234,6) (231,7) (234,6) (231,7) (234,6) (231,7) (232,7) (234,7) (232,7) (234,7) (232,7) (234,7) (232,7) (234,7) (232,7) (234,7) (232,7) (234,7) (232,7) (234,7) (232,7) (234,7) (232,7) (234,7) (232,7) (234,7) (232,7) (234,7) (232,7) (234,7) (232,7) (234,7) (232,7) (234,7) (232,7) (234,7) (232,7) (232,7) (234,7) (232,7) (232,7) (234,7) (232,7) ($

The next step would be to take out the sorted transactions. After clicking on that respective button, we get the following output on the screen.

A.L.	
Mining Itemse	High Utility ts from Transactional Databases
Preprocessing LP Grower	
Useral-e-koDeeHophives projectiransdata/Dataset-f0s1.tat	nwse Dataset 🛞 without Hadoop 😳 With Hadoop
Formaliad dataset with Transactional Utility   Disow Result	- Otean -
Enter Min Support(%) 30	Transaction weighted utility [ Botted Transactions ]
Transaction weighted utility	Transactions with Utility
3-4         - A           3-3         - 337           1-3.3         - 337           2-3.3         - 334           2-3.4         - 334           2-3.4         - 334           2-3.4         - 334           2-3.4         - 344           2-3.3         - 334           2-3.4         - 344           2-3.4         - 344           2-3.4         - 344           2-3.4         - 344           2-3.4         - 344           2-3.4         - 344           2-3.4         - 344           2-3.4         - 344           2-3.4         - 344           2-3.4         - 344           2-3.4         - 344           2-3.4         - 344           2-3.4         - 345           3-3.4         - 345	$ \begin{array}{c} T7 = (T4, T7)(T4, T4, T1(T4, T2, T1) = T4 \\ T4 = T4 = T4 = T4 = T4 \\ T5 = T4 = T4 = T4 \\ T5 = T5 = T5 = T5 \\ T5 = T5 = T5 \\ T5 = T5 = T5 \\ T5 \\ T5 = T5 \\ T5 = T5 \\ T5 \\ T5 \\ T5 = T5 \\ T5 \\ T5 = T5 \\ T5 \\ T5 = T5 \\ T5$

After this, we will generate the Up-Growth tree.

	UPGrowth Res	ult		
node	) enumeror			Databases
411	411	14		Lover training and the
515		17	15	
343	848			
2014	411.313	13	10	
331	221			
3227.9	.013	0		
12.2.4				
434	459,213	29		
334	49.8,013.0214	31		
	2223	16		Parts the Language Lifer Council
134	411.021	7.3		Participa Constanting
121	.411.213.134	34		
431	441,010	3.4		
431	411,221,434	24		
331	121	14		
.902		41		
3.95		83		
395	343,224	90		
321	.433,313,134,431,332	6.1		
324	.411.313.131.832	37		
.321		18		
148	221,534	10		
.142		2.4		
142	413,221,134,133	30		
244	411,313,221,134	345	(* )	
(ALCONG)		1000		
414 0	411110201104 2		30	
14.4.4 10	411 313 001 13 0		8.0	
447 0	411,313,221,13, 2		55	
224 9	304 3			
134 0	.023		10	
142 0	3251,134 3	14	10	
3.4.4 0	221,134,142 8		0.62	
414 0	221,134,142,344 3		30	
2233 0	2229 434 442 34		46	

#### IJARSE, Vol. No.4, Special Issue (02), February 2015

# ISSN-2319-8354(E)

The next step is to find out the number of PHUIs generated and also to calculate the time required for this whole process. The following screen shows,

and in the second second			A CONTRACTOR OF A CONTRACTOR O		
	UPGrowth Res	ult		and the second sec	
node	attual	190		Databases	and the second second
411	143.73		1.	Catabases	
240	(313)	24	5		A second s
212	213	4			
221	122.0	34		and the second se	and the second se
221	.313	11			
221	321	+			
221	.313				
221	,433,	6.			
221	,433	11			
134	(134)	85		Physics of the Contract of Management	(President)
131	(123)	65	2.1	Present Conting Softwaren	1 designation of
131	,411,313,134	37	2		
121	,411,313	11			
121	.411.134	13		whole process Time without Hador	p.72088 millierconde
131	131	14		<ul> <li>South and a state of the state</li></ul>	A THE REPORT OF A CARE OF A
332	[332]	7.4			
232	411,212,134,131	41			
032	,411,313,131	23			
-5.52	313				
303	[307]	110			
321	.411.313.134.131.232	0.1			
323	.411.515.151.352	37			
301	313,332	10			
140	[142]	7.0			
142		19			
1984 0	411 313 221 154 2		25 11		
414 0	411.313.221.13 2	1	36		
#44 0	A11.313.221.13. 2	+	52		
441 0	411.313.221.13 2	7	- 56	R	
223 1	221 3	,	4		
7.34 0	,221 3	1	10		
142 0	321,134 3		10		
344 0	,221,124,142 3	. 4	35		
414 0	221,134,142,344 3		30		
233 6	.221,134,142,34 3	*	48		

The next step is to calculate the number of PHUIs generated with Up-Growth+ and also to calculate the time required.

13				Internet State	- 0 1
		UPGrowth+ R	esult		
node		- phot	1 eur	11-11-1	Databagee
411		1411)	9	14	Databasa
313		1313	24	5	
0.01		(223)	24		
221		313			
0.0.4		411			the second se
234		11141	0.5		
100		14341	0.3		
4.94		411.104	17		
330		1999	7.4		
3.54		212	1.9		
224		1000	1.00		PHUra Using UPGrowt. Graph
100		(Jac 1)	110		
447		14.470	14		
142		200 250 254	20		Contraction of the second s
194		1.410 mile 1.404	20		whole process time without tandoon 57404 destination
197		14 1-5 (44 1) (44 1)	14		whole process rinks without handout average management
140		224 424 424		- 24	
182			2.5		
142		(11.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1	19		
244		[344]	101		
010		14142	101		
233		12.3.33	158		
241		COM 1	140		
133		(133)	257		
		(844)	140		
444		313.221,332.321		12.1	
1.4.3.1	8	311 311 221 2		31	
344	0.1	411313221134 2		36	
414	0.	411.313.221.13. 2	1	36	
444	0	411,313,221,13 2		52	
441	0.	.411.313.221.13 2		56	
221	1	221 0			
134	0	.221 3	1	10	
342	0	.221,134 3		19	
344	0.1	221,134,142 0	. 1	36	
414	0.	221,134,142,344 3	.1	39	
2/3/3	0	221,134,142,34 3	1	48	

In both of the above screens shown, were the output of PHUIs generation with Up-Growth and also Upgrowth+, but that was without Hadoop. The time required for Up-Growth is 72088 milliseconds and the time required for the whole process in Up-growth+ is 59404 milliseconds which is less than the previous one. Though the difference is not so high, but it would be of greater concern when it comes to a dataset which is of large size. The time difference would matter a lot and consequently it would also matter to the amount of resources consumed in this process.



IJARSE, Vol. No.4, Special Issue (02), February 2015

ISSN-2319-8354(E)

# IX. TESTING WITH HADOOP

In this section, we are going to calculate transaction weighted utility with Hadoop.

Mining High Utility Itemsets from Transactional Databases           Presocessing (P-Growth)           UsersW-IcoDestoputwek projectframs date/Dataset100r1M           Browse Dataset           Formation dataset with Transactional Utily           Bhow Result           Chear           Formation dataset with Transactional Utily           Bhow Result           Chear           Formation weighted utility           Transaction weighted utility           133 - 406           132 - 406           144 - 414           142 - 436           144 - 414           142 - 436           144 - 414           143 - 331           144 - 414           143 - 434           143 - 434           144 - 414           144 - 414           144 - 414           144 - 414           144 - 414           144 - 414            144 - 414           144 - 414           144 - 414           144 - 414           144 - 414           144 - 414           144 - 414           144 - 414           144 - 414           144 - 414           144 - 416<		
Presencessing         UP-Growth           UserstAI-tc:Desktopswekt project/transdutu/Dataset10x11x1         Browse Dataset1         I without Hackoop         I without Hackoop           Formated dataset with Transactional Utility         Bhow Result         Clear           Enter Min Support(%)         30         Transaction weighted utility         Sorted Transactions           133 = 446         312 = 446         Transaction swith Utility         Transactions with Utility           134 = 446         144 = 410         144 = 410         144 = 414         144 = 414         144 = 414         144 = 414         144 = 414         153 = 10223 (124 / 1133 /	Mini Item	ng High Utility sets from Transactional Databases
Users/Al-k:/Deaktopt/wet.project/trans.ddu//Dataset110it110       Browse Dataset       Owthout Hackoop       With Hadoop         Formated dataset with Transactional Utility       Bhow Result       Clear         Enter Min Support(%)       30       Transaction weighted utility       Estreed Transactions         133 = 495       32       Fransaction weighted utility       Estreed Transactions       Fransaction weighted utility         133 = 495       32       Fransaction weighted utility       States Transactions with Utility         134 = 445       32       Fransaction weighted utility       States Transactions with Utility         134 = 445       142       143       142       143       142       143       144       143       144	Preprocessing UP-Growth	
Formated dataset with Transactional Utility         Bhow Result         Clear           Enter Min Support(%)         30         Transaction weighted utility         Sorted Transactions           133 = 449         332 = 445         Transactions with Utility         Image: Sorted Transactions           134 = 445         134 = 445         153 = 1435,11(142,2)(434,7)(1544,3)(241,4)(414,1)(412,4)(134,1)(231,5)(143,3)(133,10)(314,8)(422,0)(41,1)(424,1)(414,1)(412,1)(134,1)(231,5)(143,3)(133,10)(314,8)(422,0)(41,1)(424,1)(424,1)(414,1)(412,1)(134,1)(231,5)(143,3)(133,10)(314,8)(422,0)(41,1)(424,1)(424,1)(414,1)(424,2)(221,2)(133,2)(131,2)(214,4)(331,4)(333,9)(43,11-4223,1)(424,1)(414,1)(242,5)(224,2)(223,1)(422,2)(221,2)(133,2)(214,4)(331,4)(133,3)(43,1)(124,3)(214,4)(414,2)(221,2)(133,2)(411,4)(444,2)(221,2)(212,3)(134,4)(414,2)(221,2)(133,10)(134,4)(442,2)(231,1)(134,4)(432,2)(411,1)(444,4)(212,2)(221,2)(133,2)(131,1)(122,2)(141,1)(444,4)(212,2)(214,1)(131,1)(222,5)(234,3)(233,1)(44,4)(231,2)(141,4)(422,4)(231,1)(134,4)(432,2)(211,1)(434,4)(132,5)(133,10)(134,4)(432,2)(131,1)(124,3)(133,10)(134,4)(422,4)(233,1)(134,4)(432,2)(131,1)(124,4)(132,5)(134,1)(134,4)(132,5)(134,1)(134,4)(132,5)(134,1)(134,4)(132,5)(134,1)(134,4)(132,5)(134,1)(134,4)(132,5)(134,1)(134,4)(132,5)(134,1)(134,4)(132,5)(134,1)(134,4)(132,5)(134,1)(134,4)(134,2)(134,1)(134,4)(132,5)(134,1)(134,4)(134,2)(134,1)(134,4)(132,5)(134,1)(134,4)(132,5)(134,1)(134,4)(132,5)(134,1)(134,4)(134,2)(134,1)(134,4)(134,2)(134,1)(134,4)(134,2)(134,1)(134,4)(134,2)(134,1)(134,4)(134,2)(134,1)(134,4)(134,2)(134,1)(134,4)(134,2)(134,1)(134,4)(134,2)(134,1)(134,4)(134,2)(134,1)(134,4)(134,2)(134,1)(134,4)(134,2)(134,1)(134,4)(134,2)(134,1)(134,4)(134,2)(134,1)(134,4)(134,2)(134,1)(134,4)(134,2)(134,1)(134,4)(134,2)(134,1)(134,4)(134,2)(134,1)(134,4)(134,2)(134,1)(134,4)(134,2)(134	Users/Al-kdDesktop/wwk.project/transdata/Dataset/10x1.td	Browse Dataset
Formated dataset with Transactional Utility         Bhow Result         Clear           Enter Min Support(%)         30         Transaction weighted utility         Sorted Transactions           133 = 495         332 = 495         132 = 495         154 = 123,11(142,2)(433,1)(134,2)(343,1)(321,2)(134,1)(232,5)(334,3)(144,5)(423,1)(332,1)(312,2)(41,4)(412,1)(134,1)(231,5)(143,3)(133,0)(134,8)(422,6)(231,2)(414,4)(414,1)(444,2)(332,1)(133,2)(214,4)(331,4)(333,9)(431,1)(444,42)(332,1)(313,2)(214,4)(331,4)(333,9)(431,1)(444,42)(332,1)(313,2)(214,4)(331,4)(333,9)(431,1)(444,42)(332,1)(312,2)(214,4)(331,4)(333,9)(431,1)(444,42)(332,1)(132,2)(212,2)(133,2)(414,1)(414,1)(414,2)(322,2)(212,3)(324,4)(314,1)(333,2)(431,1)(422,2)(414,4)(331,4)(333,2)(431,1)(422,2)(434,4)(414,1)(334,2)(332,2)(411,1)(444,2)(322,3)(223,3)(234,3)(133,0)(134,3)(134		
Catter Min Support(%)       30       Transaction weighted utility       Sorted Transactions         133 = 495       332 = 485       15=(333,1)(1422)(431,7)(542,1)(321,2)(144,2)(313,1)(222,5)(334,3)(314,5)(423,1)(312,2)(414,2)(412,2)(423,1)(312,2)(414,2)(413,1)(332,1)(313,0)(314,8)(422,6)(217)(422,1)(414,2)(314,1)(412,1)(134,1)(231,5)(144,3)(133,0)(314,8)(422,6)(217)(444,42)(332,1)(312,2)(214,4)(331,4)(333,9)(431,1)(444,42)(332,1)(312,2)(214,4)(331,4)(333,9)(431,1)(444,42)(332,1)(312,2)(214,4)(331,4)(333,9)(431,1)(444,42)(332,1)(312,2)(214,4)(331,4)(333,9)(431,1)(444,42)(332,1)(312,2)(214,4)(331,4)(333,9)(431,1)(444,44)(21,2)(213,1)(142,2)(212,2)(1432,2)(212,2)(132,2)(143,2)(212,2)(212,2)(132,2)(212,2)(142,2)(212,2)(1	Formated dataset with Transactional Utility Bhow R	Esuft Clear
Enter Min Support(%)         30         Transaction weighted utility         Sented Transactions           133 = 495         15=(333,1)(1422)(431,7)(542,1)(522,5)(334,3)(514,5)(423,1)(332,2)(415,2)		
Transaction weighted utility133 = 495133 = 495132 = 485144 = 420144 = 420144 = 420144 = 420144 = 420144 = 436142 = 436144 = 314143 = 319142 = 436153 = 495164 = 313174 = 1436175 = 1211,110 (122,110) (122,110) (122,110) (122,110) (122,110) (123,21) (133,21) (133,14) (133,19) (131,110) (134,21) (133,19) (131,110) (134,21) (133,19) (131,110) (134,21) (133,19) (131,110) (134,21) (132,110) (134,21) (132,21) (134,21) (133,21) (	Enter Min Support(%) 30	Transaction weighted utility Sorted Transactions
$ \begin{array}{llllllllllllllllllllllllllllllllllll$	and the second second	
$ \begin{array}{c} 15-(33,1)(142,2)(431,7)(543,1)(321,4)(442,2)(313,1)(232,5)(334,3)(314,5)(423,1)(312,2)(413,3)(332,4)(312,2)(413,3)(332,4)(333,4)(333,4)(332,4$	Transaction weighted utility	Transactions with Utility
$ \begin{array}{c} 332 = 485 \\ 321 = 485 \\ 321 = 485 \\ 321 = 485 \\ 77 - (422, 1) (431, 1) (23, 1) (131, 7) = 28 \\ 70 - (121, 4) (421, 7) (134, 4) (243, 2) (411, 9) (321, 10) (424, 1) (444, 2) (332, 1) (313, 2) (214, 4) (331, 4) (333, 9) (43) \\ 444 = 420 \\ 344 = 414 \\ 73 - (134, 6) (414, 4) (312, 5) (324, 2) (434, 1) (444, 2) (332, 1) (313, 2) (214, 4) (331, 4) (333, 9) (43) \\ 74 - (441, 1) (222, 2) (232, 2) (434, 1) (444, 2) (332, 1) (134, 9) (213, 1) (134, 9) (213, 1) (134, 2) (213, 3) (134, 4) (331, 1) (122, 3) (144, 4) (132, 5) (324, 2) (242, 4) (222, 1) (243, 3) (233, 3) (344, 4) (331, 1) (122, 3) (144, 4) (132, 5) (324, 2) (244, 5) (321, 7) (442, 2) (223, 3) (344, 4) (331, 1) (122, 3) (144, 2) (133, 1) (1222, 3) (134, 5) (422, 1) (134, 3) (133, 1) (132, 1$	133 = 495	15=1333(1)(142)2)(4517)(543)(1)(5214)(444)2)(513(1)(222)5)(3343)(314,5)(423(1)(352))(41
$ \begin{array}{c} 321 = 485 \\ 1142 = 436 \\ 144 = 420 \\ 344 = 414 \\ 414 = 414 \\ 231 = 387 \\ 231 = 387 \\ 231 = 387 \\ 231 = 337 \\ 233 = 332 \\ 233 = 339 \\ 333 = 332 \\ 243 = 331$	332 = 485	T6-+(211,11) (123,1) (131,4) (142,3) (241,4) (411,1) (412,1) (134,1) (231,6) (143,3) (133,10) (314,8) (422,6) ( T7, (423,1) (423,5) (433,5) (434,3) (422,6) (
$ \begin{array}{l} 142 = 436 \\ 171 \rightarrow (233,5) (121,7) = 22 \\ 1444 = 430 \\ 172 \rightarrow (441,2) (221,2) (133,2) (434,1) (414,1) (344,3) (213,1) (134,3) (433,2) (121,5) (435,5) = 69 \\ 173 \rightarrow (134,6) (414,4) (1312,5) (324,2) (242,4) (221,3) (422,5) (221,2) (243,3) (223,3) (344,4) (331,1) (122,3) (142,2) (142,3) (142,3) (142,3) (143,3) (133,1) (143,3) (143,1) (142,3) (143,3) (142,2) (142,3) (123,3) (143,3) (143,3) (143,3) (143,3) (144,3) (143,3) (123,3) (123,3) (143,3) (143,3) (142,2) (142,3) (123,3) (142,2) (142,2) (143,3) (123,3) (142,2) (142,2) (143,3) (123,3) (143,3) (143,3) (142,2) (143,3) (142,2) (143,3) (143,3) (142,2) (142,2) (143,3) (142,2) (142,3) (143$	321 = 485	The (122, 1) (434, 7) (131, 7) -20 The -1(2) 4) (421 7) (131, 7) -20 The -1(2) 4) (421 7) (131, 7) -20
$ \begin{array}{c} 444 = 420 \\ 344 = 441 \\ 344 = 441 \\ 344 = 444 \\ 73 - (134, 4)(444, 4)(221, 2)(313, 2)(434, 4)(444, 4)(221, 3)(143, 2)(411, 4)(433, 2)(121, 5)(413, 5) = 60 \\ 73 - (134, 6)(444, 4)(312, 5)(324, 2)(242, 4)(222, 1)(422, 5)(221, 2)(243, 3)(233, 3)(344, 4)(331, 1)(122, 3)(134, 3)(234, 4)(334, 3)(134$	142 = 436	T1-(23)5)(1217)=22
$ \begin{array}{c} 344 = 414 \\ 414 = 414 \\ 73 + (1344) (1212) (324,2) (324,2) (224,2) (221,2) (224,3) (223,3) (344,4) (31,1) (122,2) (143,2) (233,3) (133,0) (313,1) (143,3) (231$	444 = 420	T2->(441,2) (221,2) (313,2) (434,1) (414,1) (344,1) (344,8) (213,1) (134,9) (411,1) (433,2) (121,5) (413,5) =68
$ \begin{array}{c} T4 + (34, 1) \ (334, 2) \ (332, 3) \ (414, 5) \ (234, 5) \ (232, 3) \ (232, 3) \ (233, 3) \ (133, 1) \ (433, 1) \ (434, 1) \ (132, 4) \ (444, 2) \ (232, 3) \ (232, 3) \ (233, 3) \ (133, 1) \ (432, 1) \ (232, 2) \ (432, 3) \ (232, 3) \ (233, 3) \ (133, 1) \ (132, 1) \ (232, 2) \ (432, 3) \ (232, 3) \ (233, 3) \ (133, 1) \ (132, 1) \ (232, 2) \ (432, 3) \ (232, 3) \ (233, 3) \ (133, 1) \ (132, 1$	344 = 414	T3->(134,6) (414,4) (312,5) (324,2) (242,4) (223,1) (422,5) (221,2) (243,3) (233,3) (344,4) (331,1) (122,3) (14
$ \begin{array}{c} 241 = 362 \\ 241 = 362 \\ 121 = 339 \\ 422 = 336 \\ 422 = 335 \\ 422 = 335 \\ 423 = 331 \\ 423 = 331 \\ 423 = 331 \\ 423 = 331 \\ 423 = 331 \\ 423 = 331 \\ 424 = 331 \\ 424 = 335 \\ 424 = 335 \\ 425 = 335 \\ 425 = 335 \\ 425 = 335 \\ 426 = 335 \\ 426 = 335 \\ 426 = 335 \\ 427 = 335 \\ 427 = 335 \\ 428 = 331 \\ 438 = 331 \\ 438 = 331 \\ 438 = 331 \\ 438 = 331 \\ 438 = 331 \\ 438 = 331 \\ 438 = 331 \\ 74 = (434, 4)((434, 2)((332, 4)(444, 3)(444, 4))(444, 6)((23, 3)((334, 3)(411, 1)(4332, 6)(434, 5)(413, 5)(132, 6)(334, 6)(411, 1)(433, 2)(123, 5)(444, 5)(321, 6)(331, 6)(331, 6)(313, 6)(431, 6)(432, 6)(321, 6)(321, 6)(331, 6)(313, 6)(322, 6)(331, 6)(313, 6)(313, 6)(313, 6)(313, 6)(313, 6)(313, 6)(313, 6)(313, 6)(313, 6)(322, 6)(331, 6)(313$	231 = 387	T4+-(344,11) (334,2) (332,3) (411,1) (414,5) (244,5) (321,7) (442,3) (232,3) (233,3) (133,8) (313,1) (443,1) (1
$ \begin{array}{c} 125 = 339 \\ 123 = 339 \\ 223 = 339 \\ 223 = 335 \\ 335 = 331 \\ 242 = 335 \\ 335 = 331 \\ 243 = 331 \\ 243 = 331 \\ 335 = 331 \\ 243 = 331 \\ 243 = 331 \\ 335 = 331 \\ 243 = 331 \\ 335 = 331 \\ 243 = 331 \\ 335 = 332 \\ 243 = 331 \\ 336 = 332 \\ 336 = 332 \\ 336 = 331 \\ 336 = 331 \\ 336 = 331 \\ 336 = 331 \\ 336 = 331 \\ 336 = 331 \\ 336 = 331 \\ 336 = 331 \\ 336 = 331 \\ 336 = 331 \\ 336 = 331 \\ 336 = 331 \\ 336 = 331 \\ 336 = 331 \\ 336 = 331 \\ 336 = 331 \\ 336 = 331 \\ 336 = 331 \\ 336 = 331 \\ 336 = 332 \\ 336 = 336 $	241 = 362	T5->(333,1) (142,2) (431,7) (343,1) (321,4) (444,2) (313,1) (232,5) (334,3) (314,5) (423,1) (332,1) (312,2) (41
$ \begin{array}{c} 233 = 339 \\ 17^{-(422,1)}(424,1)^{-(134,4)}(131,4)^{-(144,2)}(32,1)0(424,1)(444,2)(332,1)(31,2)(214,4)(331,4)(333,0)(43) \\ 422 = 335 \\ 422 = 335 \\ 33^{+} = 331 \\ 243 = 331 \\ 243 = 331 \\ 33 = 331 \\ 243 = 331 \\ 33 = 312 \\ 243 = 319 \\ 33 = 312 \\ 223 = 300 \\ 33 = 312 \\ 223 = 300 \\ 316 = 312 \\ 316 = 312 \\ 336 = 31$	121 = 339	16-4(21,11) (125,1) (131,4) (142,3) (241,4) (411,1) (612,1) (134,1) (231,5) (143,3) (133,30) (314,3) (422,6) (
$\begin{array}{c} 422 = 335 \\ 71 = (223,5)(122,7)(1=3) \\ 243 = 331 \\ 243 = 331 \\ 243 = 331 \\ 233 = 334 \\ 233 = 334 \\ 233 = 331 \\ 233 = 331 \\ 233 = 331 \\ 233 = 331 \\ 233 = 331 \\ 233 = 332 \\ 233 = 332 \\ 233 = 332 \\ 233 = 332 \\ 233 = 332 \\ 233 = 332 \\ 233 = 332 \\ 74 = (134,6)(144,2)(132,2)(322,3)(144,4)(131,1)(122,5)(234,2)(142,2)(233,3)(244,4)(331,1)(122,2)(433,3)(333,4)(331,1)(443,5)(1333,3)(333,4)(333,$	233 = 339	17 -> (422, 1) (434, 7) (131, 7) -> 17 Th> (423, 1) (434, 7) (334, 7) (443, 7) (443, 7) (351, 10) (454, 4) (735, 3) (733, 7) (734, 4) (735, 4) (735, 10) (451, 10) (735, 10) (451, 10) (735, 10) (451, 10) (735, 10) (451, 10) (735, 10) (451, 10) (735, 10) (451, 10) (735, 10) (451, 10) (735, 10) (451, 10) (735, 10) (451, 10) (735, 10) (451, 10) (735, 10) (451, 10) (735, 10) (451, 10) (735, 10) (451, 10) (735, 10) (451, 10) (735, 10) (451, 10) (735, 10) (451, 10) (735, 10) (451, 10) (735, 10) (451, 10) (735, 10)
$\begin{array}{c} 331 = 334 \\ 243 = 331 \\ 72 - (4412)(2212)(3132)(4341)(414,4)(243,1)(444,8)(213,1)(134,9)(411,1)(433,2)(121,5)(413,5)=100 \\ 73 - (1346)(141,4)(212,5)(224,2)(224,2)(224,2)(224,3)(223,3)(233,3)(244,4)(331,3)(142,3)(412,3)(232,3)(233,3)(233,3)(233,3)(143,3$	422 = 335	10**12331(****)(*****)(******)(*****)(*****)(******
$\begin{array}{c} 243 = 331 \\ 73 \sim (134,6) \ (414,4) \ (312,5) \ (324,2) \ (222,3) \ (422,6) \ (221,2) \ (243,3) \ (233,3) \ (344,4) \ (331,1) \ (122,3) \ (142,3) \ (333 = 312 \\ 233 = 312 \\ 233 = 312 \\ 233 = 300 \\ 243 = 2300 \\ 314 = 303 \\ 314 = $	331 = 331	72-(441,2)(221,2)(313,2)(434,1)(414,1)(344,1)(444,8)(213,1)(134,9)(411,1)(433,2)(121,5)(413,5)=10
$ \begin{array}{c} 144 = 319 \\ 133 = 312 \\ 223 = 300 \\ 314 = 303 \\ 143 = 303 \\ 143 = 303 \\ 144 = 303 \\ 143 = 303 $	243 = 331	T3~(134.6) (414.4) (312.5) (324.2) (242.4) (223.1) (422.5) (221.2) (243.3) (233.3) (344.4) (331.1) (122.3) (14
75->(333,1) (142,2) (431,7) (343,1) (321,4) (444,2) (313,1) (232,5) (334,3) (314,5) (423,1) (332,1) (312,2) (413           76-(211,11) (123,1) (131,4) (142,3) (241,4) (411,1) (412,1) (134,1) (231,5) (143,3) (133,10) (314,8) (422,6) (2           743 = 303	1434 = 319	T4-+(344,11) (334,2) (332,3) (411,1) (414,5) (244,5) (321,7) (442,3) (232,3) (233,3) (133,8) (313,1) (443,1) (1
314 = 303 143 = 303 143 = 303	223 = 300	T5-+(333,1) (142,2) (431,7) (343,1) (321,4) (444,2) (313,1) (232,5) (334,3) (314,5) (423,1) (332,1) (312,2) (41
143 = 300	314 = 303	16-(211,11)(123,1)(131,4)(142,3)(241,4)(411,1)(412,1)(134,1)(231,5)(143,3)(133,10)(314,8)(422,6)(
	143 = 303	MERCANDESCRIPTION (19)

Next step is to find out the sorted transactions.

4	
Preprocessing UP-Growth	g High Utility ets from Transactional Databases
Usersivi-kollesktopwivek projectifansidataillataseti tukt, tit	drowse Dataset
Exempled dataset with Transactional Liblity Show Result	Cierc
( Torring on the second count ) ( Count ( and	
Fater Min Compared(0/)	
Later Min Support(76) 30	Transaction weighted usiny Sorted transactions
Transaction weighted utility	Transactions with Utility
L'ACRAME	
133 = 495	Transactions -> Itemsets = RTU
321 = 485	10-1411(3)(313,2)(134,4)(131,0)(332,1)(321,10)(444,2)(231,1)(331,4)(243,2)=90
142 = 436	12-(411.1)(313.2)(221.2)(134.9)(344.1)(414.1)(444.8)(441.2)=56
444 = 420	T3->(221,2) (134,6) (142,1) (344,4) (414,4) (233,3) (223,1) (422,5) (323,2) (312,5) (331,1) (243,3) =112
344 = 414	T4->(411,1) (313,1) (131,3) (332,3) (321,7) (344,11) (414,5) (233,3) (241,1) (133,8) (223,4) (334,2) (232,3) (44
414 = 414	T5->(313,1) (221,1) (332,1) (321,4) (142,2) (444,2) (334,3) (232,5) (314,5) (143,1) (323,7) (312,2) =104
231 = 387	T6->(411,1) (221,5) (134,1) (131,4) (142,3) (241,4) (133,10) (422,6) (314,8) (143,3) (231,5) =144
121 = 339	(1/2(13), 1/(422, 1) = 10 Transations themesh =
233 = 339	TB-state 9/134 4/ (313.2)/131 8/ (32.1)/321 10/ (444.2) (231.1)/121 4/ (331.4)/243.2) (434.5)/333.9) = 1
422 = 335	T1(121,7)(233,5)=33
331 = 331	T2->(411,1) (134,9) (313,2) (221,2) (444,8) (344,1) (414,1) (121,5) (434,1) =77
243 = 331	T3->(134,6) (221,2) (142,1) (344,4) (414,4) (233,3) (422,5) (331,1) (243,3) (223,1) =82
434=319	14->(411,1) (313,1) (131,3) (133,8) (332,3) (321,7) (344,11) (414,5) (241,1) (233,3) (223,4) =109
223 - 306	15 - 1333, 11(227, 1)(332, 1)(327, 4)(142, 2)(444, 2)(333, 1)(314, 3)(143, 1)=54 TE - (44, 5)(45, 4)(45, 4)(45, 4)(45, 4)(45, 2)(45, 5)(45, 4)(45, 5)(44, 5)(45, 4)(45, 5)(44, 5)(45,
314 = 303	172-(1317)(42,1)(131,4)(251,3)(133,10)(142,3)(231,3)(241,4)(422,5)(314,8)(143,3)=118
143 = 303	

The next step is to observe the PHUIs generation with UP-Growth which is shown in the screen below.

IJARSE, Vol. No.4, Special Issue (02), February 2015

# ISSN-2319-8354(E)

	UPGrowth	Result		Databases
node	phui	nu		
411	{411}	18		
134	{134}	30		
134	134	12		
313	(313)	32		
313	,411,	24		
313	.411	3		
313	313	1		PHUR Using LPGrowt Grant
131	(131)	56		There could be brown.
221	(221)	63		
221	,411,134,	24		
221	,134	16		whole process Time without Hadoop 72088 millseconds
221	,411,134,131	18		whole process Time without Hadoop 59404 miliseconds
225	,411,134,313	26		whole process Time with Hadoop 31525 milliseconds
221	,313	3		
221	,411,134,	13		
133	[133]	42		
133	,411,,131	6		
133	,411,134,131,221	28		
133	,411,313,131	14		
133	,411,,131,221	6		
133	,411,134,131,	8		
332	(332)	76		
332	,411,134,313,131	39		
332	,411,313,131,	26		
332	.313,221	8		
0	411 134 313 221 2		50	
0	411.134.313.22 2	1	54	
0	.411.134.313.22 2	1	55	
D	411,134,313,22 2	1	75	
0	411, 134, 313, 22 2	1	77	
1	134 3	1	12	
0	134 3	1	16	

The next step is to observe the PHUIs generation with UP-Growth+.

4					_ = ×
	UPGrowth+	Result		THE R. L.	
node	phui	nu		Databases	
411	(411)	18		Databases	and the second se
134	(134)	38	5	Contract Contract Contract	
313	13431	12			
313	411	16			
131	[131]	55			
221	/2241	63			
221	411 134	25			
221	411 134	12			
133	(133)	42			
133	411 131	13		Contractor and the second seco	E contract 1
122	411 121 221	16		PHUEs Using UPGrowt	Graph
133	411 124 121	25			
332	13321	75			
332	,411,313,131, .313, .211	15		whole process Time without Hadoo	72088 miliseconds
222				whole process Time without Hadon	adoop 59404 millseconds
321		118		whole process Time with Hadoop 31525 milliseconds	
325	313 332	13			
147	(142)	84		whole process Time with Hadoop 18437 milliseconds	8437 milliseconds
142	221 332 321	23			
142	313 221 321	16			
147	313 221 332	8			
444	(444)	145			
444	313 221 332 324	10			
344	1344)	177			
414	(414)	187	1		
333 0	411 134 313 13 0		152		
121 1	121 1	1	28		
233 0	.121 1	1	33		
221 0	.411.134.313 2	1	26		
444 0	.411,134,313,221 2	1	50	2	
344 0	.411,134,313.22 2	1	54		
414 0	.411.134.313.22. 2	1	55		
121 0	.411.134.313.22. 2	1	75		
434 0	411.134.313.22 2	1	77		
134 1	134 3	1	12		
	474		44		

From the above screens, we can see that with hadoop data mining is very efficient as compared to data mining without hadoop. As you can see, the time required for the whole process for Up-Growth with Hadoop is 31525 milliseconds and the time required for the whole process for Up-Growth+ with Hadoop is 18437 milliseconds which is the least.

So we can conclude from these outputs that the time required for the whole process for Up-Growth without Hadoop is the highest whereas the time required for the whole process for Up-Growth+ with Hadoop is the least.

Following screen shows the graph which compares the amount of PHUIs generated in Up-Growth and in Up-Growth+.

IJARSE, Vol. No.4, Special Issue (02), February 2015

# ISSN-2319-8354(E)



Following screen shows the full comparison of the performance i.e. Up-Growth and Up-Growth+ with and without Hadoop.

		M Ite	ining h msets	tigh Util Train Tr	ity chsaction	al Databases	
Preprocessio	D UP-Growth						
LIP-TH	ne filosoft	UP-Growth	PHOJI's Lise	ig tot-throwth	UP-Growth+	PHUra Using UPDowet	Orapti
tement	Plant	Posteon	Ptow	Long	PALALINE		
11		411	0		10	<ul> <li>whole process Time without Hade</li> </ul>	op 59404 millseconds
34	0	.411	0	3	26	whole process firms with Hadoop	31505 milleeconds
13	0	.411.134	9	2	28		
34	0	.411,134,313	0	4	34	whole process Time with Hadoost	16437 millisscomili
32	0	ALCONDER THE			30	100	
21	0	Christian Christian			Tra-arts	TITLE - CONTRACTOR	
4.4	0			Granh			
0.4	0	S 400 000		car apri			
21	D	8 100 100					
4.5	0	Br (100,000)					
2.4	- A	C 600.000					
13		4 800,000					
2.4		000.000					
0.0	0	G 100 mm					
11	n	E 100.000 1					
44	n .	100.000	100		10000		
4.4	0	1 2 0		-			
14	-	- E	and the second	-	- and the second s		
215			14		A 194		
34	0			Altered	relation .		
34	1		10				

# X. CONCLUSION

As shown above, the time required for the processing of data in Hadoop with UP-Growth+ is very less which proves that it is very efficient as compared to the earlier papers.

# REFERENCES

- [1] Vincent S. Tseng, Bai-En Shie, Cheng-Wei Wu, and Philip S. Yu, Fellow, IEEE, "Efficient Algorithms for Mining High Utility Itemsets from Transactional Databases", IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 25, NO. 8, AUGUST 2013.
- [2] V incent . S. Tseng, C. W. Wu, B. E. Shie, and P. S. Yu.: UP Growth : An Efficient Algorithm for High Utility Itemset Mining. In Proc. of ACM-KDD, Washington, DC, USA, pp. 253- 262, July 25– 28, 2010.
- [3] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules," Proc. 20th Int'l Conf. Very Large Data Bases (VLDB), pp. 487-499, 1994.

International Journal of Advance Research In Science And Engineering

http://www.ijarse.com

IJARSE, Vol. No.4, Special Issue (02), February 2015

ISSN-2319-8354(E)

- [4] 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. Knowledge and Data Eng., vol. 21, no. 12, pp. 1708-1721, Dec. 2009.
- [5] C.H. Cai, A.W.C. Fu, C.H. Cheng, and W.W. Kwong, "Mining Association Rules with Weighted Items," Proc. Int'l Database Eng. and Applications Symp. (IDEAS '98), pp. 68-77, 1998.
- [6] J. Han, J. Pei, and Y. Yin, "Mining Frequent Patterns without Candidate Generation," Proc. ACM-SIGMOD Int'l Conf. Management of Data, pp. 1-12, 2000.
- [7] S.C. Lee, J. Paik, J. Ok, I. Song, and U.M. Kim, "Efficient Mining of User Behaviors by Temporal Mobile Access Patterns," Int'l J. Computer Science Security, vol. 7, no. 2, pp. 285-291, 2007.
- [8] H.F. Li, H.Y. Huang, Y.C. Chen, Y.J. Liu, and S.Y. Lee, "Fast and Memory Efficient Mining of High Utility Itemsets in Data Streams," Proc. IEEE Eighth Int'l Conf. on Data Mining, pp. 881-886, 2008.
- [9] Y.-C. Li, J.-S. Yeh, and C.-C. Chang, "Isolated Items Discarding Strategy for Discovering High Utility Itemsets," Data and Knowledge Eng., vol. 64, no. 1, pp. 198-217, Jan. 2008.
- [10] C.H. Lin, D.Y. Chiu, Y.H. Wu, and A.L.P. Chen, "Mining Frequent Itemsets from Data Streams with a Time-Sensitive Sliding Window," Proc. SIAM Int'l Conf. Data Mining (SDM '05), 2005.
- [11] Y. Liu, W. Liao, and A. Choudhary, "A Fast High Utility Itemsets Mining Algorithm," Proc. Utility-Based Data Mining Workshop, 2005.
- [12] F. Tao, F. Murtagh, and M. Farid, "Weighted Association Rule Mining Using Weighted Support and Significance Framework," Proc. ACM SIGKDD Conf. Knowledge Discovery and Data Mining (KDD '03), pp. 661-666, 2003.
- [13] J. Han and Y. Fu, "Discovery of Multiple-Level Association Rules from Large Databases," Proc. 21th Int'l Conf. Very Large Data Bases, pp. 420-431, Sept. 1995.
- [14] J. Han, J. Pei, and Y. Yin, "Mining Frequent Patterns without Candidate Generation," Proc. ACM-SIGMOD Int'l Conf. Management of Data, pp. 1-12, 2000.
- [15] S.C. Lee, J. Paik, J. Ok, I. Song, and U.M. Kim, "Efficient Mining of User Behaviors by Temporal Mobile Access Patterns," Int'l J. Computer Science Security, vol. 7, no. 2, pp. 285-291, 2007.
- [16] H.F. Li, H.Y. Huang, Y.C. Chen, Y.J. Liu, and S.Y. Lee, "Fast and Memory Efficient Mining of High Utility Itemsets in Data Streams," Proc. IEEE Eighth Int'l Conf. on Data Mining, pp. 881-886, 2008.
- [17] Y.-C. Li, J.-S. Yeh, and C.-C. Chang, "Isolated Items Discarding Strategy for Discovering High Utility Itemsets," Data and Knowledge Eng., vol. 64, no. 1, pp. 198-217, Jan. 2008.
- [18] C.H. Lin, D.Y. Chiu, Y.H. Wu, and A.L.P. Chen, "Mining Frequent Itemsets from Data Streams with a Time-Sensitive Sliding Window," Proc. SIAM Int'l Conf. Data Mining (SDM '05), 2005.
- [19] Y. Liu, W. Liao, and A. Choudhary, "A Fast High Utility Itemsets Mining Algorithm," Proc. Utility-Based Data Mining Workshop, 2005.
- [20] R. Martinez, N. Pasquier, and C. Pasquier, "GenMiner: Mining nonredundant Association Rules from Integrated Gene Expression Data and Annotations," Bioinformatics, vol. 24, pp. 2643-2644, 2008.