A Novel Technique to find Frequent Items by using Greedy Association Rule Mining

[I]V. Supriya, [II]P R Sudha Rani
[I]Dept. of CSE, Shri Vishnu Engineering College For Women, Vishnupur, Bhimavaram
[II]Assoc. Prof, Dept. of CSE, Shri Vishnu Engineering College For Women, Vishnupur, Bhimavaram

Abstract

Frequent Itemset Mining is one of the established data mining issues in the majority of the data mining applications. It requires substantial computations and I/O movement limit. Additionally assets like single processor's memory and CPU are extremely constrained, which corrupts the execution of algorithm. In this paper we have proposed one such dispersed algorithm which will keep running on Hadoop – one of the current most famous appropriated structures which for the most part concentrate on map reduce worldview. The proposed approach considers inborn attributes of the Apriori algorithm identified with the continuous itemset era and through a piece based apportioning utilizes a dynamic workload administration. The algorithm significantly upgrades the execution and accomplishes high versatility contrasted with the current disseminated Apriori based methodologies. Proposed algorithm is executed and tried on vast scale datasets distributed over a group.

Keywords

Apriori Algorithm, Map Reduce, Frequent Itemset Mining, Hadoop, Distributed Computing.

I. Introduction

Data mining is the powerful procedure of finding designs which are already obscure and covered up in expansive datasets. Current improvements and advances in many developing territories of designing, science, business, and so forth are creating gigantic measure of data step by step bringing about overwhelming prerequisite of capacity. The effectiveness to process, dissect, and comprehend these datasets is at the need of a few orders, including parallel and circulated figuring. This is because of their innate dispersed nature, the nature of their substance, the extent of the datasets and the heterogeneity and so forth. A standout amongst the most critical zones of data mining is association administers mining; it is an assignment is to discover all items or subsets of items which habitually happen and the connection between them by utilizing two primary strides: finding frequent itemsets and creating association rules. Frequent Itemset Mining (FIM) tries to discover information from database in perspective of consistent occasions of an event as showed by the base repeat edge gave by customer. In context of destinations of basic memory, FIM bends up without a doubt inefficient on immense databases. This issue can be settled by utilizing Apriori figuring, where database is isolated diverse conditions for rehash check of each size of sure itemsets. Shockingly, single machines can't satisfy the memory necessities for dealing with the entire strategy of contender itemsets. Likewise existing algorithms care to control the yield and runtime by expanding the base recurrence edge, consequently diminishing the quantity of competitor and continuous itemsets. Parallel writing computer programs is getting most extraordinary importance to deal with the enormous measures of information, which is made and used every day. Parallel programming models and supporting calculations, can be accumulated into two major orders viz. shared memory and circled (share nothing). On shared memory frameworks, all preparing units can in the meantime get to an ordinary memory zone. While, scattered frameworks are made out of processors that have their own particular inside recollections and chat with each other by passing messages. It is simpler to port algorithms to shared memory parallelism; however they are regularly not sufficiently adaptable. Appropriated frameworks, allow semi coordinate flexibility for particularly balanced activities. Nonetheless, it is not generally simple to compose or even adjust the projects for disseminated systems. Current algorithms like Apriori are useful for the databases that are little in estimate, yet in the event that these algorithms are executed on substantial databases in parallel on conveyed systems the execution can be enhanced fundamentally. Hadoop is an open source appropriated system which is planned in view of the Google's Map-diminish programming model. Hadoop is fit for dissecting substantial measure of data. Hadoop is produced by remembering the vast majority of the items like-huge dataset, compose once read many access models, moving algorithm is less expensive than moving data and so on. Apache Hadoop wins terabyte sort benchmark in July 2008. This capacity makes Hadoop appropriate for most mining issues. Hadoop has its own document system called Hadoop Distributed File system (HDFS) which is equipped for running on product equipment with high adaptation to non-critical failure capacity. Data replication is one of the vital highlights of HDFS, which guarantees data accessibility and programmed re-execution on different node disappointment. In this paper we have proposed algorithm which will utilize the energy of Hadoop for mining the Frequent Itemset.

II. Related Work

Frequent itemsets is thought to be essential in numerous data mining undertakings that endeavor to find intriguing examples from databases, for example, connections, association rules, scenes, successions, groups, classifiers and numerous increasingly where association govern mining is the most well known issue. The first incitement of enthusiasm for seeking association rules originated from the need of detail examination of alleged general store exchange data, i.e. to consider client conduct regarding the bought items. Association rules tells how as often as possible the items are obtained together. For instance, an association manages (bread) - > (eggs) (80%) states that four out of five clients that purchased bread likewise purchased eggs. Such guidelines can be matter of significance for choices about store format, item valuing, advancements and numerous others. With the presentation of algorithm by Agrawal et al. since 1993, the issue of mining the frequent itemset and association run are thought to be of most extreme essential. Inside the previous decade, various research papers have been distributed exhibiting novel algorithms or changes on existing algorithms to take care of these mining issues all the more proficiently. Numerous variations and enhancements of this algorithm have been produced reasonable in parallel and disseminated frameworks, for example, CD, FDM.

The Apriori Algorithm AIS algorithm by Agrawal et al. was the main algorithm which creates all continuous itemsets and sure association rules with presentation of this mining issue. Agrawal et al. enhanced an indistinguishable algorithm and renamed it from Apriori which makes utilization of monotonicity property of the help of itemsets and the certainty of association rules. Apriori algorithm is an exemplary algorithm for finding continuous itemsets which is for the most part in light of level insightful hunt and iteratively find frequent itemsets with measure from 1 to k-itemset. Essential thought is to limit the hunt space by utilizing the Apriori standard:

An itemset must be Frequent if and just if the majority of its subsets are Frequent.

That is, if {AB} is an Frequent itemset, at that point both {A} and {B} ought to be Frequent

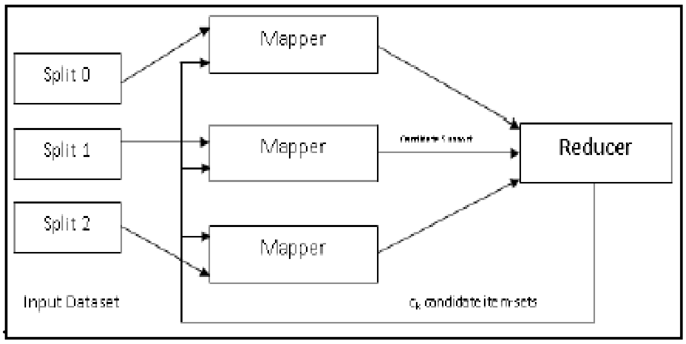


Fig. 1: Data flow layout between mapper and reducer of classic

Apriori algorithm In the event that there are n 1-itemsets that fulfill your base help, Apriori and numerous different algorithms must consider n*(n-1)/2-itemsets. This obviously gets rather costly. In Apriori, the 2-itemsets regularly is the biggest and most costly stride and 3-itemsets might be more regrettable.

A Frequent-Pattern Tree Approach Mining regular examples in time-arrangement databases, exchange databases, and different sorts of databases has been contemplated and investigated prevalently in data mining research. In addition, applicant set era is still exceptionally costly, particularly when we manage huge number of examples and/or long examples. J. Pei, J. Han, and Y. Yin had proposed a novel successive example tree (FP-tree) structure. This is a broadened prefix-tree structure utilized for putting away packed and critical data about continuous examples. It utilizes an effective FP-development mining approach which is FP tree based, concentrating on the idea of example section development for the entire arrangement of regular examples.

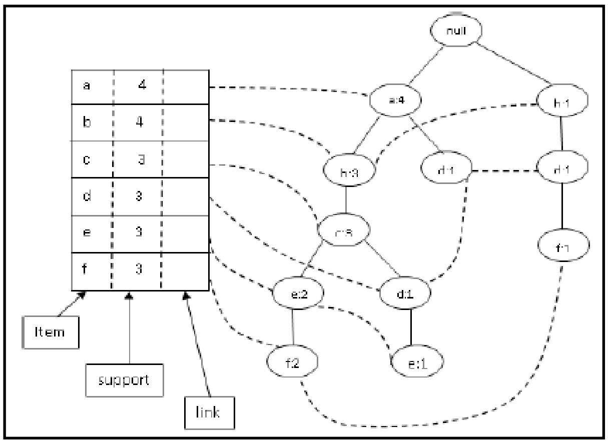


Fig. 2: An Example of FP-Tree

The fundamental preferred standpoint of FP-development is that each connected rundown, beginning from a thing in the header table speaking to the front of that thing, is put away in a packed frame. Shockingly, to finish this pick up, it needs to keep up an unpredictable data structure and play out a considerable measure of dereferencing and furthermore the FP-tree portrayal is regularly significantly bigger.

Sampling The sampling algorithm, proposed by Toivonen, performs at most two sweeps through the database by picking an arbitrary example from the database, at that point discovering all generally Frequent designs in that specimen, and afterward confirming the outcomes with whatever is left of the database. In the situations where the examining strategy does not create every single successive example, the missing examples can be found by producing all residual conceivably Frequent designs and checking their backings amid a moment go through the database. By reducing the help edge, the likelihood of such a slip-up can be kept away from. In any case, for a sensibly little likelihood of thwarted expectation, the edge must be surely decreased, which can cause a combinatorial effect of the measure of hopeful designs.

Apportioning

The Partition algorithm, Savasere et al. was proposed utilizes an approach which is totally not quite the same as all past methodologies. Database is put away in primary memory utilizing the vertical database design and the help of an itemset is processed by meeting the fronts of two of its subsets. To register the help of an applicant k itemset I, which is produced by combing two of its subsets X, Y as in the Apriori algorithm, it crosses the fronts of X and Y , bringing about the front of I. Unmistakably, securing the fronts of all things genuinely induces that the total database is analyzed into manage memory. For colossal databases, this could be unfathomable. Along these lines, the Partition figuring utilizes the running with trap. The database is parceled into a few disjoint parts and the algorithm creates for each part all itemsets that are moderately frequent inside that part, utilizing the algorithm depicted in the past passage and appeared in Algorithm. The parts of the database are picked such that each part fits into primary memory on itself. The algorithm blends all moderately Frequent itemsets of each part together. This outcomes in a superset of all Frequent itemsets over the entire database, since an itemset that is Frequent in the total database must be generally Frequent in one of the parts. At that point, the real backings of all itemsets are figured amid a moment look over the database. Once more, every part is perused into principle memory utilizing the vertical database format and the help of each itemset is processed by meeting the fronts of all items happening in that itemset.

III. Approach Description

This segment characterizes the issue definition, proposed algorithm, framework and execution and gives a logical dialog that portrays the proposed approach.

Problem Definition

The mathematical presentation of the essential idea of help include apriori algorithm exhibited. Let I = {i1, i2, i3,… ..,im} be the arrangement of items. Let T = {t1, t2, t3,… .,tn} be the arrangement of exchanges, where every exchange t is an arrangement of items with the end goal that t ⊆ I. The thing X has bolster s in the exchange set T is s% of exchanges contain X signified as s = support(X). An association run can be characterized as A→B, where {A, B} ⊆ I and A ∩ B = Ø. The help of administer A→B is bolster (A ∪ B). The issue of mining association lead is to discover every one of the tenets that fulfill a client indicated least help limit. The itemset X is said to be Frequent if its help is more prominent than or equivalent to the client characterized least help limit and furthermore the greater part of its subsets are additionally Frequent. Square deliberation approach is extremely useful for an appropriated document framework. To begin with, for vast records, it is not required that the pieces from a document to be put away on a similar plate, so they can exploit any of the circles in the group. Indeed, it is conceivable to store a solitary record on a HDFS bunch whose squares filled every one of the circles in the group. Second, for more improved subsystem a square can be considered as a unit of deliberation rather than a record. The capacity administration is additionally streamlined since pieces are of settled size, in this way dispenses with metadata concerns. Additionally, to provide adaptation to non-critical failure and accessibility, pieces fit well with replication. Here we consider the square parceling for the dissemination of the datasets among all preparing nodes. The dataset W is separated among M nodes with D exchanges as {T1, T2, . . . , TM}. Here each square constitutes exchanges that are allotted to the nodes. How about we expect size of the parcel Ti as Di. Presently each parcel Ti is isolated into bi squares {t1,t2,… ,tbi }. The measure of a square ti is characterized as a default estimation of 64MB or as per the accessible memory in the handling node Ni and number of items, the normal exchange width, and furthermore the help limit of a dataset. For a given least help limit delta, an itemset x is all inclusive Frequent on the off chance that it is Frequent in W; its help x.support is more noteworthy than delta × D, and is locally Frequent in a node Ni in the event that it is Frequent in Ti; its help x.support is more prominent than delta × Di.

Proposed System

As Hadoop requires data as key-value pairs as info and yield, it have to first mastermind the value-based data into a reasonable configuration where key is the exchange ID or balance of each line and qualities will be the comma isolated rundown of items in that exchange. A basic two stages and approximately coupled design can be utilized to execute expansive scope of data mining issue. Additionally for each stage in Map-Reduce, the data ought to be as key and esteem combines so we have to choose the transitional key esteem structure. These middle of the road key esteem sets are passed to the diminish stage toward the finish of the guide stage to extricate the recurrence tally of itemsets. Essentially, the algorithm comprises of two stages. The principal stage comprises of the mix era which can be viewed as principle steps. Every principle step comprises of a count on the neighborhood dataset squares, or a got obstructs in the development of nCk blends, and the potential correspondence trades for the workload and solicitations administration.

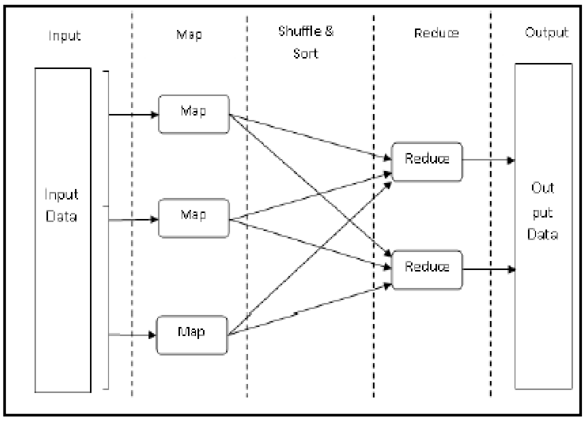


Fig. 3: Map-Reduce Dataflow

The second stage comprises of a synchronization which is completed toward the finish of the primary stage for conclusive outcomes collection. In the start of the primary stage, every node Ni is appointed a square of its dataset parcel Di, its number of pieces bi, and a workload vector Vi. This data is utilized to decide a remote execution time when another preparing node exchanges a vocation demand to some other node.

System Layout According to past discussion, number of items in the dataset and the help edge influences the algorithm many-sided quality of the Apriori era. Unmistakably the mixes of applicant sets can exponentially develop in the Apriori era process bringing about popularity in memory space. This gives rise either to a whipping impact, which can amazingly corrupt the execution, or unfit to manage the dataset if the usage is not adjusted to out of center algorithms.

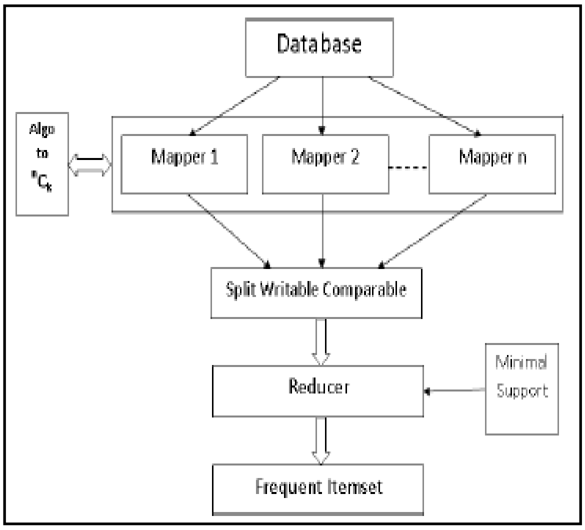


Fig 4: Proposed system layout

Every node performs then the nCk mix era in its squares autonomously of the others. In the event that a given node completes its pieces preparing, it chooses a processor that has not issued its end warning yet. This approach accomplishes workload adjust and quick execution time.

The proposed algorithm depends on Hadoop Map-Reduce programming. The database is sorted out in such an organization, to the point that each line contains data for a specific exchange. In the proposed framework is utilizing TextInputFormat as Input arrange so every mapper will get every exchange as data. On accepting items in exchange the guide stage will make mix of all items in that exchange to decrease the database check. For each such blend produced in delineate will radiate the key as thing mix and incentive as TransactionId. The yield of all guide stage will be given to the rearranging and arranging stage. To think about the thing pair blend, we have composed an altered comparator. Every reducer will get key as thing pair and incentive as rundown of all exchanges in which that thing pair happen. For each such key esteem match the reducer will ascertain the whole of all exchange in which that thing pair happen and contrast it and negligible help and discharge the yield thing pair as key and incentive as invalid. Along these lines we get frequent thing sets with less database filter yet it will expand the quantity of in-memory algorithm to produce blend. Following algorithm delineates the mapper and reducer utilized for Apriori Algorithm.

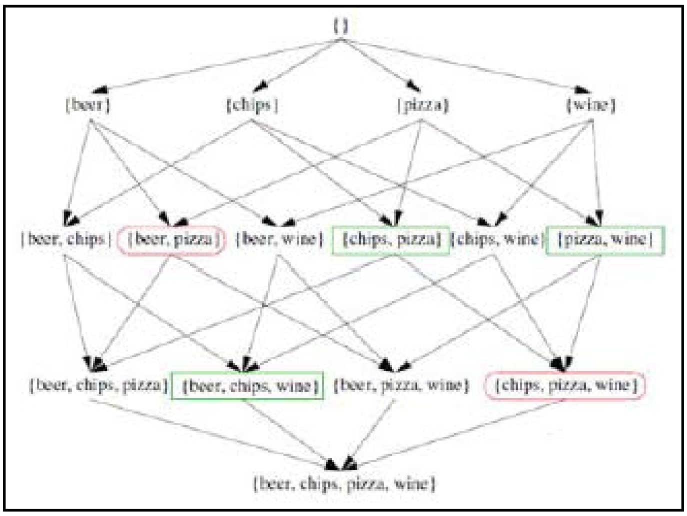


Fig 5: Lattice of the itemsets

IV. Proposed Algorithm

The proposed algorithm is an essential disseminated usage of the Apriori algorithm and it is anything but difficult to adjust to delineate. Algorithm 1 and 2 demonstrates the pseudo codes of the fitting mapper and reducer. Figure 3 gives the outline of the correspondence of the algorithm.

Input:-
D-Dataset containing different transactions with Itemsets
S- Minimal Support

Algorithm 1:- Mapper of the proposed algorithm

```
Map<Transaction_id, Itemset>
{
//Split itemset based on space
String items [] =itemset.split (— ‖)
for (int k: items.length)
{
//Generate combinations to reduce database
//scan
item_combinations=generateCombinations(n,k)
}
for(item_combination:item_combinations)
{
//For each combination emits combination as key
//and txn_id as value
emit(item_combination,txn_id)
}
}
```

In Map function of the proposed algorithm each line of the given dataset is dealt with as single exchange and is doled out an exceptional exchange ID as txnid. The itemsets in an exchange are part in light of space. On the off chance that any exchange has more than 9 items then that specific exchange is additionally part into sub exchange with subtxnid to accelerate the procedure. Subsequent stage is to frame every conceivable mix for each exchange. Development of nCk mixes decreases the database checks. Each mix is checked for symmetric property to maintain a strategic distance from copy keys. The yield of mapper is as key esteem match and is give as input o the reducer.

Algorithm 2:- Reducer of the proposed algorithm

```
Reduce<item_combination,Iterable<txnids>
{
int count=0;
//Count the occurrence of each txnid in which
//item_combination occurs
For( txnid:txnids)
{
count++;
}
If(count>=minimum_support)
{
//if count>=minimum_support emit
//item_combination as key, and Value will be null
Emit(item_combination,NullWritable);
}
}
```

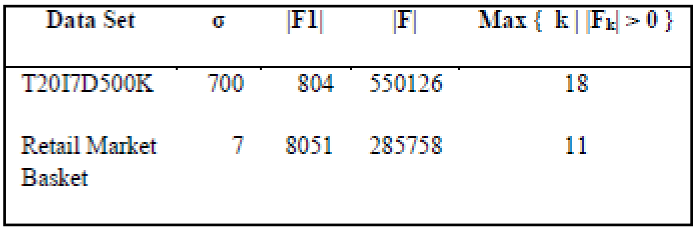
Input to the reducer is key esteem combine where key is itemset {2,3} and esteem is as <1,1,1> i.e. number of events of that itemset. The reducer just includes the qualities from key esteem match and check for whether it fulfills the base help. In the event that yes then that specific itemset is viewed as Frequent and written in yield record.

V. Dataset

The experiment is finished utilizing two genuine dataal collections which are freely accessible and have distinctive attributes. One of which is produced by IBM manufactured data generator and other is the crate dataal index, contains exchanges from a retail location. Table I demonstrates the quantity of items and the quantity of exchanges in every datum set, and the base, most extreme and normal length of the exchanges. Moreover, Table II appears for every datum set the least negligible help limit that was utilized as a part of examinations, the quantity of Frequent items and itemsets, and the extent of the longest successive itemset that was found.

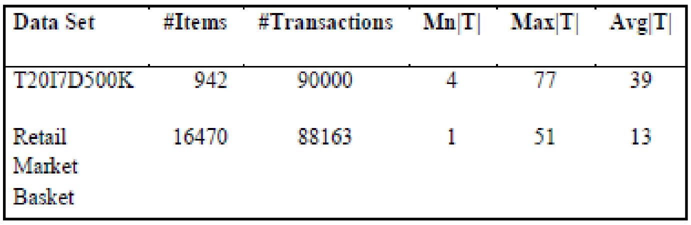
Data Set Characteristics

Table I : Number of Items & Transactions in each Data Set



Data Set	σ	|F1|	|F|	Max { k | |F_k| > 0 }
T20I7D500K	700	804	550126	18
Retail Market Basket	7	8051	285758	11

Table II : Lowest Min Threshold for each Data Set



Data Set	#Items	#Transactions	Mn|T|	Max|T|	Avg|T|
T20I7D500K	942	90000	4	77	39
Retail Market Basket	16470	88163	1	51	13

VI. Conclusions

In this paper, presented another guide map-reduce based algorithm tending to issue of mining regular itemsets using dynamic workload administration through a block-based apportioning. The piece based approach manages memory imperatives since the fundamental errand of producing mixes may require vast memory space contingent upon a few parameters including the help limit. Our approach likewise abuses an intrinsic property of the itemsets era assignment that demonstrates that the middle of the road correspondence ventures, in traditional usage, for example, the FIM approach, are execution obliging. To be sure, worldwide pruning methodologies don't acquire enough helpful data correlation with the produced synchronization and I/O overheads. The highlights of proposed algorithm are it utilizes an appropriately tuned estimation to quantify the right itemset amid the pass. It fuses administration of support and guarantees culmination. The excess is disposed of by taking care of copies painstakingly and the workload administration. It demonstrates that the proposed algorithm accomplishes great execution and high adaptability contrasted with an established Apriori-based usage.

References

[1] *Lin, Ming-Yen and Lee, Pei-Yu and Hsueh, Sue-Chen Apriori-based frequent itemset mining algorithm on Mapreduce, ICUIMC'12 Proceeding of the 6th International Conference on Ubiquitous Data Management and Communication, 2012.*

[2] *Sandy Moens, Emin Aksehirli and Bart Goethals, Frequent Itemset Mining for Big Data, Universiteit Antwerpen, Belgium.*

[3] *R.C. Agarwal, C.C. Aggarwal, and V.V.V. Prasad. Depth first generation of long patterns. In Ramakrishnan et al.*

[4] *R. Agrawal, T. Imielinski, and A.N. Swami. Mining association rules between sets of items in large databases. In P. Buneman and S. Jajodia, editors, Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, volume 22(2) of SIGMOD Record, pages 207–216. ACM Press, 1993.*

[5] *R. Agrawal and R. Srikant. Quest Synthetic Data Generator. IBM Almaden Research Center, San Jose, California, http://www.almaden. ibm.com/cs/quest/syndata.html. 38*

[6] *A. Amir, R. Feldman, and R. Kashi. A new and versatile method for association generation. Data Systems, 2:333–347, 1997.*

[7] *R.J. Bayardo, Jr. Efficiently mining long patterns from databases. In L.M. Haas and A. Tiwary, editors, Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data, volume 27(2) of SIGMOD Record, pages 85–93. ACM Press, 1998.*

[8] *C.L. Blake and C.J. Merz. UCI Repository of machine learning databases.*

[9] *Ferenc Kovacs and Janos Illes Frequent Itemset Mining on Hadoop, ICCC 2013 IEEE 9th International conference on Computational Cybrnetics, Tihany, Hungary, July 8-0, 2013.*

[10] *J. Han, J. Pei, and Y. yin. Mining Frequent Pattern Without Candidate Generation. A frequent-Pattern Tree Approach. Data Mining and Knowledge Discovery, 2003*

[11] *R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In J.B. Bocca, M. Jarke, and C. Zaniolo, editors, Proceedings 20th Inter- national Conference on Very Large Data Bases, pages 487–499. Morgan Kaufmann, 1994.*

[12] *R. Agrawal, T. Imielinski, and A.N. Swami. Mining association rules between sets of items in large databases. In P. Buneman and S. Jajodia, editors, Proceedings of the 1993 ACM SIGMOD International Confer- ence on Management of Data, volume 22(2) of SIGMOD Record, pages 207–216. ACM Press, 1993*

[13] *R.C. Agarwal, C.C. Aggarwal, and V.V.V. Prasad. A tree projection algorithm for generation of frequent itemsets. Journal of Parallel and Distributed Computing, 61(3):350–371, March 2001.*

[14] *R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A.I. Verkamo. Fast discovery of association rules. In U.M. Fayyad, G. Piatetsky- Shapiro, P. Smyth, and R. Uthurusamy, editors, Advances in Knowledge Discovery and Data Mining, pages 307–328. MIT Press, 1996*

[15] *Bart Goethals. Survey on Frequent Pattern Mining, HIIT Basic Research Unit Department of Computer Science University of Helsinki P.O. box 26, FIN-00014 Helsinki Finland.*

Authors Profile

__V. Supriya__ is currently pursuing her M.Tech (CSE) in Computer Science and Engineering Department, Shri Vishnu Engineering College For Women,West Godavari, A.P. She received her B.Tech in Computer Science and Engineering Department from Shri Vishnu Engineering College For Women, Bhimavaram.

__Mrs.P R Sudha Rani__ is currently working as an Associate Professor in Computer Science and Engineering, Department, Shri Vishnu Engineering College For Women, West Godavari.