Optimized Partitioning Based Genetic Algorithm For Generating Mining Frequent Patterns From Big Data Sets

Chandaka Babi, M.Venkateswara Rao, V.Venkateswara Rao, Bhanuja Arketla

***Abstract*: *The Data Mining developments have been built-up and investigated in terms of technologies and methodologies. Mining of Frequent patterns is unique precise data excavating tasks, mostly from merchandizing data. Our job aims to find out all chronological patterns through a customer précised minimum threshold support, and here the support of a pattern can be defines as the total of occurrences of data in the given pattern. This paper concentrates on problems associated with frequent data mining for knowledge based system. A thorough analysis has been done on these problems and answers have been made for the problems related to previous process and new techniques have been created for mining frequent patterns. Initially this research work focus on previous activities in the area of frequent pattern mining and then this research work initially proposed an algorithm Apriori with optimization using genetic Algorithm for finding the frequent patterns. Usually the running time of the procedure to discover or invent frequent items pattern based on total no of candidates produced at every level and the time consumed to read the data set. The proposed method reduces the scanning time and also reduces the number of candidate itemsets generated at each step. This is because the database can be read only for once, at that moment an intermediate dataset can be constructed at each step. Here also then the association rule generated by the Apriori algorithm is optimized using genetic algorithm. To produce strong association rules, the algorithm uses Genetic Algorithm operators like selection, crossover and mutation on association rule produced by Apriori algorithm. The parallel algorithm has been proposed to mine the frequent patterns with a user specified minimum support. The job is distributed among n number of processors to compute frequent item sets. So there will be communication between the processors. The time required to complete the job is very less when compared to other algorithms. The key disadvantage of this procedure is execution time, since number of processors used will be increased when the number of data items increased. To build it more competent, partition algorithm have been designed, in this a separate partitioning is created for each sets of data items. To get the count of a particular item sets, scanning the entire database is not required and only require the particular partition. Consequently the scan time has been decreased. Over all the algorithms, partition algorithm will have improved performance over the present algorithms*.

***Index Terms: Big Data, Data Mining, Frequent Pattern Mining, Frequent Item Sets Hybrid Apriori, Association Rules, Portioning Algorithm, and parallel execution*.**

Revised Manuscript Received on May 10, 2019

Chandaka Babi, Dept of IT, Gitam University, Visakhapatnam, India.
Dr M.Venkateswara Rao, of IT, Gitam University, Visakhapatnam, India.
Dr V.Venkateswara Rao, Dept of CSE, Sri Vasavi Engineering College, Tadepalligudem, India.
Bhanuja Arketla, CSC, Gitam University, Visakhapatnam, India

I. INTRODUCTION

Data extraction is a process that reads huge datasets with an aim to produce novel information or considering the goal knowledge finding that can be used for decision making. In general, data mining comprises fields such as classification, clustering, frequent item set and outlier detection [1, 2, 3].Mining of Frequent Patterns can be treated as one of the handy data mining approaches [4] which are in demand today and it finds features (items) that come together frequently. Various present applications that uses data mining and analysis are discovery of drugs, analysis of share markets data to find trends and market basket analysis that gives strategies for marketing. [5]. numerous algorithms have been designed for mining of Frequent patterns in the history of data mining like Apriori [6], Eclat [7], FP-Growth [8, 9, 10]. Finding all frequent patterns is a difficult job and is also emerges as high cost computational process. This is because a huge number of candidate item sets and consequently numerous procedures become in competent once it reaches large amount of data that we access during the generation process and this called as Big Data. This we can find in Social networking sites like Facebook creates more than 400TB of information every day [11]. Similarly Wal-Mart and Amazon and other big companies' record huge number of business data operations each year. Moreover DNA sequencing organizations can yield terabytes of information. To manage this difficult of huge dimensions of data, investigators planned numerous parallel procedures for problem of Frequent Pattern Mining.

In this paper, the improved Apriori algorithm called as Portioning Apriori with Optimization feature using Genetic Algorithm has been proposed and it is compared with Apriori, Apriori using Genetic algorithm. The algorithms are used on massive large data sets and it has been established that partitioned Apriori is more efficient than Apriori algorithm, Apriori using Genetic algorithm.

II. RELATED WORK

A. Introduction

One of the significant inclinations in information technology is discovering important information from

massive volume of data stored in files, databases, and storage places and to provide potent ways for analysis and construal of such data for the mining of interesting knowledge that may possibly facilitate decision-making process.

The two fold foremost motives for expending data mining are, consuming very little evidence with excessive data and the other fact to dig out helpful data from massive quantity of data. The concept of data extraction assists us to find out consequential new association, prototypes and potentially constructive knowledge from large amount of data by means of mathematical and also statistical methods. The idea of data mining has reached the industry from so many years, nevertheless the word data mining came into existence from the year 1990. Data Analysis has departed over many exploration and developmental phases used for long period of time, and for the reason that of 3diverseancestors, the concept took place at its shape and individuals ancestors are statistics, artificial intelligence and machine learning. Data extraction is constructed on statistics which is the substance of the majority tools, these includes standard variance, regression analysis, standard distribution, clusters analysis so on. Artificial intelligence is also the pedestal for data mining, which tries to simulate human thought process or human intelligence in statistical problems. Another core area for data mining is machine learning, and it is a combination of statistics and AI. Data extraction is fundamentally the alteration of machine learning practices to business requests. Data mining is compilation of past and topical changes in statistics, Artificial Intelligence, and machine learning. These techniques are aimed to observe and find unseen patterns otherwise knowledge available in data.

. ***Frequent Pattern Mining Evolution***

Frequent item sets are sub sets, which can be extracted from data set that persuades customer specific minimum support value with occurrence that is below a user specific onset. Researchers projected a proficient procedure that produces complete association rules among the items in the consumer transaction data set and this procedure comprises new appraisal, pruning methods and supervision of buffer [12]. Later Apriori and Apriori Tid are designed for solving the problem of extracting frequent patterns, and then combining these two algorithms. This technique effects in producing much lesser number of candidate items sets. The AprioriTid algorithm contains a character that the data set is not employed at all for calculating the support of candidate item sets once we complete the first pass.

Later by enhancing existing association rule mining methods, a top down tolerant expanding method is presented for extracting multiple level association rules from huge transaction data sets. There after people developed DBMiner a relational data mining system for extracting multiple classes of rules at different concept stages, like association rules, characteristic rules, classification rules, discriminant rules. Later we have graph centered algorithm to extract the frequent items in the data set and this procedure constructs correlation graph to indicate the relations between items, and then traverse the graph to generate large item sets and sequences[14]. To discover generalized sequential patterns, a new algorithm called GSP was developed[24].

.

B. Improved Frequent Pattern Mining Techniques

As an improvement to above algorithms FP-growth was proposed, for mining the complete list of frequent item sets, the effectiveness of extraction can berealizedusing3methods: a huge database is densed into a extremely reduced much lesser data structure, that escapes expensive, recurrent data set reads, FP Tree based extraction embraces a pattern splinter progress technique to evade the expensive creation of a enormous number of candidate sets. Cheng-Yue Chang et al.,[11] explore a new algorithm named Segmented Progressive Filter which segment the data set into sub-data sets in such a way *that* items in each sub-data set will have either the common starting period or common ending period. For every sub-data set, Segmented Policy Filter progressively filters candidate 2itemsets by means of cumulative filtering thresholds whichever forward or backward in period. This feature allows SPF of adopting the scan reduction technique by generating all candidate k-itemsets (k >2) from candidate 2itemsets directly. William Cheung et al [84], elites CATs tree extended the idea of FP- Tree to improve the storage compression then extraction of frequent patterns without generating the candidate item sets and allow single pass over the data set and insertion and deletion of transactions in the tree can be done very efficiently. Chung-Ching Yu et al., [16] suggested2 efficient methods for discovering sequential patterns from d-dimensional sequence data. First algorithm AprioriMD algorithm is an updated version of Apriori to extract sequential patterns from a multi-dimensional data set. It employs tree structure called candidate tree similar to hash tree. The other algorithm is prefix span method which is the updated version of prefix span method. The classical Apriori and AprioriTid algorithm, are used to construct the frequent itemset, the main disadvantage is that it consumes more time for scanning the database. In order to avoid this, Zhi-Chao Li et al [97]generates a highly efficient AprioriTid algorithm, for reducing the scanning time. The improved algorithm by Claudio Lucchese et al., [17] embraces a specific visit and partitioning approach of the search space, constructed on a novel hypothetical framework that optimizes space and time.

.

C. Mining Frequent Patterns using Genetic Algorithm

Researchers proposed a genetic algorithm technique to remove the weak item sets to provide the continued existence of the best ones by running on the large population. By means of genetic based approaches of having fitness function, best association rule is also discovered. It is more efficient than Apriori algorithm, and the time take for generating the association rule is very less, so that efficiency is increased. Rule created by Association Rule Mining method does not reflect the negative incidences of attributes in them. Nevertheless with the help of Genetic Algorithms, the system can forecast the rules that contain negative attributes.

III. MINING FREQUENT PATTERNS USING PORTIONING ALGORITHM

The concept of Partitioning was first introduced in Oracle data base technology 8.0 and it is the most essential and fruitful functional aspect of the Oracle database. Partition is a method to divide data sets, tables, indices, and tables organized by index into reduced fragments called partitions and that permits the data set items to be achieved and retrieved at a greater level of scale. Every partition is identified by its exact name and takes its individual features such as its storage and index. To illustrate partitioning methodology, suppose a Human Resource manager has one big box that encompasses employee files. Each file stipulates the employee joining date. Queries are executed frequently for employees joined in a specific month. One answer is to build an index on employee joining date that stipulates the locations of the files dispersed all over the box. Another solution is partitioning approach. This employs numerous reduced boxes where each box comprising files for employees joined in a specified month. Use of reduced size boxes has numerous advantages. To extract the files for employees joined in August, the manager can extract the Auguest box. If any small box is momentarily scratched, the additional reduced size boxes can be used. Partitioning is essential, as soon as data set size is larger than 2 gigabytes, data sets encompass past data and the insides of a table need to be dispersed diagonally diverse kinds of storage devices. Partitioning is a tool for erection of multi peta byte systems with great high accessibility necessities. This process is effective in time complexity since it permits the maintenance and failure operation on a specific partition to be conceded out on designated partitions whereas other partitions are accessible to users so that the process increases the enactment of job. The independence nature of partition permits for simultaneous use of the numerous partitions for several purposes. The main advantage is that it reduces the total cost of data ownership by maintaining the old relevant information in a compressed format and improves the performance, manageability, reduced contention for shared resources and increases the availability. Partitioning is beneficial for numerous different kinds of application that control huge bulks of data. From the viewpoint of a database administrator, a segregated object has numerous pieces that can be controlled individually or jointly and have flexibility in managing a partitioned item.

A. Partitioning Strategies

Data base or data set Partitioning provides 3essential data dispersal methods as elementary partitioning approaches that regulate by what means data is located into discrete partitions. They are range, hash and list. Data set can be partitioned conferring to the data dispersal methods asa single-level partitioning and complex level partitioning. The single level partitioning practices only one of process of data distribution like range or hash or list in one or extra columns as the partitioning key.

The composite partitioning is a grouping of two data dispersal methods which that defines a composite partition table. Initially, a data set is partitioned by one data dispersal technique, and then each partition is again divided into sub partitions with the help of another data dispersal technique. Partitions of a composite partitioned table can be termed as meta-data, and it will not denote the definite data storage.

1. Range Partitioning

In range partitioning, established on a range of values of the partitioning key-value the data is dispersed. Ranges are continually well-defined as and is counting upper boundary of a partition, and the lower boundary of a partition is inevitably well-defined by the elite upper boundary of the previous partition and partition margins are always cumulative. Each partition takes data smaller than condition that identifies an upper-bound for the partitions.

2. List Partitioning

In list partitioning, data set practices a set of distinct data items as partition key-value to every partition. List partitioning is employed to regulate how specific rows draw to specific partitions, and can group the associated lists of data once the key-value used to recognize them is not suitably well-ordered. List partitioning, specifies, for a region column as the partitioning key, the 'India' partition might comprise values 'Chennai', 'Mumbai' and 'Delhi'. To hook completely data for a partition key which are not unambiguously well-defined by some of the lists, a special default partition can be defined.

3. Hash Partitioning

In hash partitioning, hash algorithm in memory is used to the partitioning key-value for determining the partition for an assumed partition key. Hash function does not offer any rational drawing between the data and some other partition; nevertheless it offers equally balanced sizes of the partitions. This is the main advantage of hash partitioning. The hashing procedure is calculated to dispense rows evenly across devices so that each partition comprises about the similar number of rows. Hash partitioning is valuable for separating large data sets into several small pieces to increase manageability. The composite partition techniques are further divided into following categories

1) range hash 2) range-list 3) range-range 4) list-range 5) list-list 6) list-hash

7) hash-hash 8) hash-range 9) and hash-list.

B. Mining Frequent Item sets using Partitioning Algorithm

A partition idea has been planned to surge the running time speed with lowest cost. For every item set, i.e1- item set, 2-itemset, 3-itemsetetc., a distinct partition will be formed in data insertion in to the data set. Firstly a set of frequent 1-item sets are found by reading the data set and fetch the numbers of

incidences of every item from the partition of those specific items using the pointer, the items fulfilling the minimum support count are encompassed in the frequent 1-itemset L_1. Like Apriori algorithm L_1aim to discover L_2, the set of L_2then used to discover L_3, and so on, until no additional frequent k-i tem sets are found. To get L_k, it is not essential to read the whole data set; it is sufficient to pursuit the tally of each data item set from its partition. Initially to generate frequent item set, a significant property termed as the Apriori property used for decreasing the search space. Join and Prune are 2 steps to get the frequent item sets. In join operation, L_k is discovery from a list of candidate k-item sets C_k which is created by connecting$_{Lk-1}$ with itself.

In pruning, to discover the count of every candidate in C_k, the partition of every item-set would be tested and the count that is not smaller than lowest support count are frequent and belongs to L_k. To reduce the size of C_k, the Apriori property is used. The performance of partition algorithm for finding the frequent data items is efficient when compared to other existing Algorithms.

Table 1 Transaction Database for Partition Algorithm

TID	List of Items
TR1	O1, O2, O4
TR2	O2, O4
TR3	O3, O4
TR4	O1, O2, O5
TR5	O1, O4
TR6	O1, O3, O5
TR7	O1, O2,O3, O5
TR8	O1, O2, O3

Consider the Data set D, having seven transactions, now partitioning has been employed to store and retrieve the data from the data set. Different types of partitioning techniques are available, in this thesis; range partitioning has been used to increase the performance when extracting the frequent patterns in the data set.

Table 3.1, represents a transaction data set for partition algorithm and it comprises 8 transactions. In following table, transaction T1 holds I1,I2,I3 and transaction T2 holds I2, I4 and so on..using candidate 1 item set C1 which produces the frequency 1 item set L1 with the help of calculating the no of occurrences of the data items directly from the partition instead of reading entire data set again. Here the minimum support count taken is2. Candidate 1 item set which is meeting the minimum support count and that is included in L1. The figure 3.1.shows, the generation of candidate 1itemsets and frequent 1 item sets.

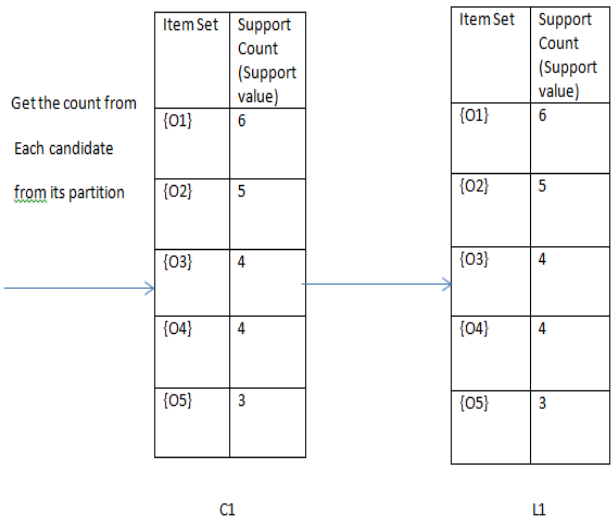

Figure 3.1 the generation of candidate 1 item sets and frequent 1 item sets.

The candidate 2 item sets are created by union of L1 with itself and verify whether the subset of the frequent item sets are also frequent, since in L1 the total items have been encompassed, here we have no pruning. For computing the support value, in place of reading the entire data set, it is sufficient to fetch the count from the suitable partition. The candidate 2 item sets that are meeting the minimum support value will be compassed in the frequent 2 item sets L2. The figure 3.2 shows the creation of frequent 2 item sets with the help of partition. At last 8 frequent 2 item sets have been created, and it has been used to produce the candidate 2 item sets.

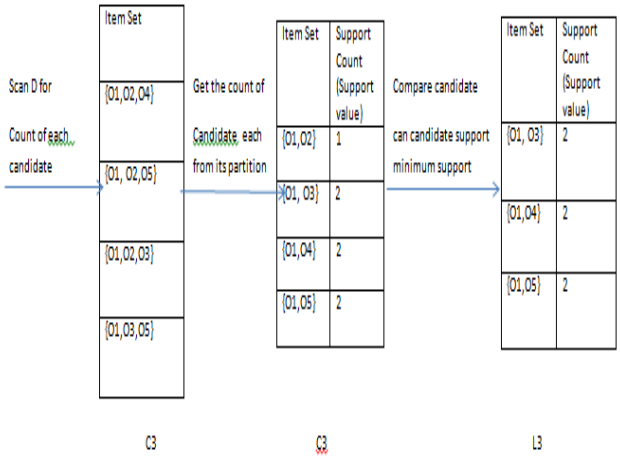

Figure 3.2 Generation of Frequent 2- Item set using Partition Algorithm

The candidate 3 item sets are created by union of L2 by itself and discover whether the subset of the frequent item sets are again frequent, only the item sets {O1,O2,O4},{O1,O2,O3},

{O1,O2,O5},and {O1,O3,O5} have been incorporated for the following stage which is identified as candidate 3 item sets because its subset is also a frequent item set. The residual item sets have been detached because its subset is not identified as frequent. To estimate the support value in its place of reading the entire data set, it is sufficient to get the count from the suitable partition. The candidate 3 item sets that are sustaining the minimum support value will be encompassed in the frequent 3 item sets L3. The figure 3.3 displays how frequent 3 item sets are produced with the help of partition. lastly 3 frequent 3 item sets can be produced, and it has been used to generate the candidate 3 item sets.

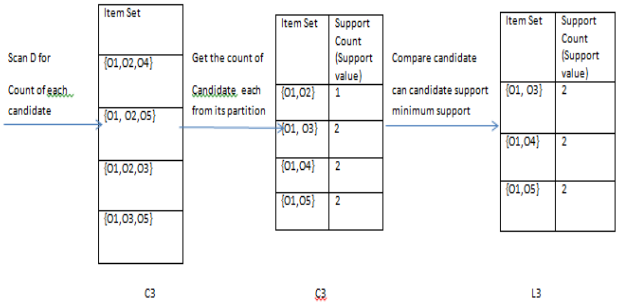

Figure 3.3 Generation of Frequent 3 – Item set using Partition Algorithm

The candidate 4 item sets are produced through combining L3 with itself, then verify the subset of the frequent patterns are frequent also, simply when we included the item sets {O1,O2,O3,O5}d for the next step then that is identified as candidate 4 item sets, because it is subset whose subset is again a frequent item set. For calculating the support count, directly get the count from the partition, instead of scanning entire data set. Since the support count is 1, which is not adequate the minimum support, it is pruned and L4 = ¢, finally process terminates. Through the frequent 3 item sets, association rule is pragmatic that displays the relation between the data items.

IV. EXPERIMENTAL EVALUATION

The partition algorithm has been implemented using Big Data sets and then compared with Apriori algorithm. The following shows Performance of Partitioning algorithm.

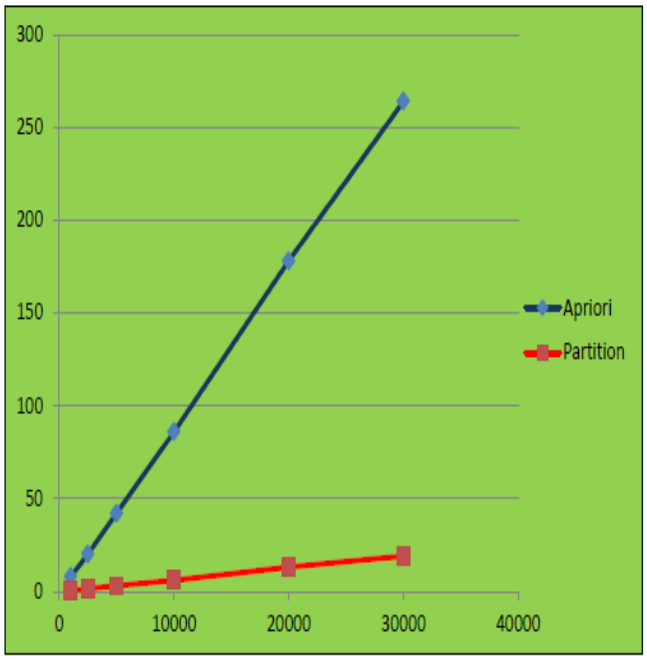

Figure 4.1 Performance evaluation of Partitioning algorithm

The Figure 4.1 explain the how frequent item sets are generated and association rule generation graph and by using the table 6.2, graph have been plotted for Apriori and partition algorithms. In this graph, total no of records has taken in the x-axis, and the execution time has taken as y-axis. Graphical representation shows, partition algorithm has efficient performance over the Apriori algorithm.

Here 10000 Scale means 10 Lakh, 20000 means 20 Lakh son on

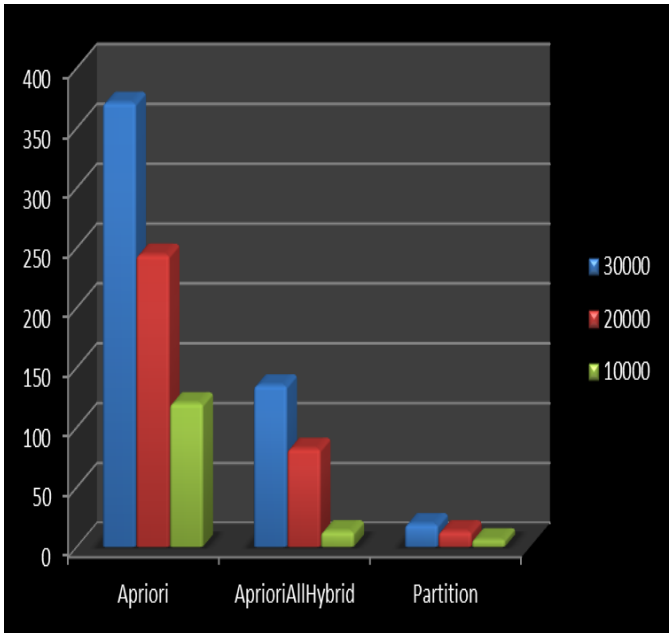

Figure 4.2 Comparison of Apriori, Apriori Hybrid and Partition algorithms

The running time is increasing as number of customer transactions increases in the data set. When compared to other algorithms, the partition algorithm has taken less running time for finding the frequent patterns and association rules by considering the items of transactions in data set.

V. CONCLUSION

This paper addresses the necessity of finding frequent patterns in the data mining. Problems related to frequent pattern mining have been analyzed and found a better solution for it. Basically the Apriori Association Rule Mining algorithm is employed to find the frequent patterns available in the database. Initially the work starts with the proposed AprioriHybrid algorithm for mining a frequent pattern which performs better when compared to Apriori algorithm and it has been proved by taking various size of data set. The association rule generated by Apriori algorithm is optimized using genetic algorithm and then a parallel algorithm has been proposed which is efficient and proved with some sample data, but the disadvantage is cost. Finally partition algorithm have been proposed, in this approach control will move to the particular partition instead of scanning the entire data set. This paper concludes that the projected partition algorithm method time and again performs effectively than other algorithms and will support various upcoming researches in numerous approaches.

REFERENCES

1 Cheikh TidianeDieng, Tao-Yuan Jen, Dominique Laurent, and Nicolas Spyratos.Mining frequent conjunctive queries using functional and inclusion dependencies. VLDB J., 22(2):125 150, 2013.

2 Bart Goethals, Dominique Laurent, Wim Le Page, and Cheikh TidianeDieng.Mining frequent conjunctive queries in relational databases through dependency discovery.Knowl. Inf. Syst., 33(3):655 684, 2012.

3 Cheikh TidianeDieng, Tao-Yuan Jen, and Dominique Laurent.An effcientcomputation of frequent queries in a star schema. In Database and Expert Systems Applications, 21th International Conference, DEXA 2010, Bilbao, Spain, August 30 - September 3, 2010, Proceedings, Part II, pages 225 239, 2010.

4 Kun He, Yiwei Sun, David Bindel, John E. Hopcroft, and Yixuan Li. Detecting overlapping communities from local spectral subspaces. In *2015 IEEE International Conference on Data Mining (ICDM 2015), Atlantic City, NJ, USA*, pages 769-774, 2015.

5 Ayad Ibrahim, Hai Jin, Ali A. Yassin, and DeqingZou.Towards privacy preserving mining over distributed cloud databases. In *Proceedings of the 2nd International Conference on Cloud and Green Computing (CGC 2012), Xiangtan, Hunan, China*, pages 130{136. IEEE Computer Society, 2012.

6 Leila Ismail and Liren Zhang.Modeling and performance analysis to predict the behavior of a divisible load application in a cloud computing environment.*Algorithms*, 5(2):289{303, 2012.

7 Fan Jiang and Carson Kai-Sang Leung. Stream mining of frequent patterns from delayed batches of uncertain data. In *Proceedings of the 15th International Conference on Data Warehousing and Knowledge Discovery (DaWaK 2013), Prague, Czech Republic*, pages 209{221. Springer-Verlag New York, Inc., 2013.

8 Alfredo Cuzzocrea, LadjelBellatreche, and Il-Yeol Song. Data warehousing and olap over big data: Current challenges and future research directions. In *Proceedings of the 16th International Workshop on Data Ware-housing and OLAP (DOLAP 2013), San Francisco, California, USA*, pages 67{70. ACM, 2013.

9 MaluCastellanos, Chetan Gupta, Song Wang, and UmeshwarDayal.Leveraging web streams for contractual situational awareness in operational BI. In *Proceedings of the 2010 International Conference on Extending Database Technology/International Conference on Database Theory (EDBT/ICDT 2010) Workshops, Lausanne, Switzerland*, pages7:18. ACM, 2010.

10 Alfredo Cuzzocrea, Carson Kai-Sang Leung, and Richard Kyle MacKinnon. Mining constrained frequent itemsets from distributed uncertain data. *Future Generation Computer Systems*, 37:117{126, 2014.

11 Alfredo Cuzzocrea, DomenicoSaccia, and Jefrerey D. Ullman. Big data: A research agenda. In *Proceedings of the 17th International Database Engineering & Applications Symposium (IDEAS 2013), Barcelona, Spain*,pages 198{203. ACM, 2013.

12 Alfredo Cuzzocrea. CAMS: OLAPing multidimensional data streams efficiently. In *Proceedings of the 11th International Conference on Data Warehousing and Knowledge Discovery (DaWaK 2009), Linz, Austria*,pages 48-62. Springer verlag, 2009.

13 Yifan Chen, Xiang Zhao, Xuemin Lin, and Yang Wang.Towards frequent subgraph mining on single large uncertain graphs. In *2015 IEEE International Conference on Data Mining (ICDM 2015), Atlantic City,NJ, USA*, pages 41{50, 2015.

14 Jeferey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Commun. ACM*, 51(1):107{113, 2008.

15 [dsonDela Cruz, Carson Kai-Sang Leung, and Fan Jiang. Mining `following' patterns from big sparse social networks. In *Proceedings of the International Symposium on Foundations and Applications of Big Data Analytics (FAB 2016), San Francisco, CA, USA*, pages 923{930. ACM,2016.

16 Mohammad El-Hajj and Osmar R. Zaiane. Parallel bifold: Largescale parallel pattern mining with constraints. *Distributed and Parallel Databases*, 20(3):225{243, 2006.

17 Mohammad El-Hajj and Osmar R. Parallel leap: Large-scale maximal pattern mining in a distributed environment. In *Proceedings of the 12th International Conference on Parallel and Distributed Systems (ICPADS 2006), Minneapolis, USA*, pages 135{142. IEEE, 2006.

18 Fan Jiang, Carson Kai-Sang Leung, Dacheng Liu, and Aaron M. Peddle. Discovery of really popular friends from social networks. In *Proceed- ings of the 4th IEEE International Conference on Big Data and Cloud Computing (BDCloud 2014), Sydney, Australia*, pages 342{349, 2014.

19 Dongme Sun, ShaohuaTeng, Wei Zhang, Haibin Zhu, "An Algorithm to Improve the Effectiveness of Apriori", In Proc. Int"l Conf. on 6th IEEE Int"l Conf. on Cognitive Informatics (ICCI'07), 2007.

20 Mannila, H. and Toivonen, H.,"Discovering generalized episodes using minimal occurrences", In Proc. of ACM Conference on Knowledge Discovery and Data Mining (SIGKDD), Pages 146–151, 1996.

21 Anandhavalli M, Suraj Kumar Sudhanshu, Ayush Kumar and Ghose M.K. (2009) Optimized Association Rule Mining using Genetic Algorithm, Advances in Information Mining, ISSN:0975-3265, Volume 1, Issue 2, 2009, pp-01-04.

22 Markus Hegland. The Apriori Algorithm – a Tutorial, CMA, Australian National University, WSPC/Lecture Notes Series, 22-27, March 30, 2005.

23 Bart Goethals, Dominique Laurent, Wim Le Page, and Cheikh TidianeDieng.Mining frequent conjunctive queries in relational databases through dependency discovery.Knowl. Inf. Syst., 33(3):655 684, 2012.

24 Woo, J., Xu, Y, "Market Basket Analysis Algorithm with Map/Reduce of Cloud Computing", In Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications, 2001.

25 Lin, Ming-Yen, Pei-Yu Lee, Sue-Chen Hsueh, "Apriori-based Frequent Itemset Mining Algorithms on MapReduce", In Proceedings of the 6th International Conference on Ubiquitous Information Management and Communication, ACM, 2012.

26. Mannila H, Toivonen H. Discoveringgeneralized episodes using minimaloccurrences. In Proc. of ACM Conferenceon Knowledge Discovery and Data Mining(SIGKDD). 1996;146–151.

27. Li H, Wang Y, Zhang D, Zhang M, ChangEY. Pfp: Parallel fp-growth for queryrecommendation. In Proc. RecSys.2008;107–114.

28. Lin MY, Lee PY, Hsueh SC. Apriori-basedfrequent Itemset mining algorithms onMapReduce. In Proc. ICUIMC, ACM. 2012;26–30.

29. Malek M, Kadima H. Searching frequentitemsets by clustering data: Towards aparallel approach using mapreduce. InProc. WISE 2011 and 2012 Workshops,Springer Berlin Heidelberg. 2013;251–258.

30. Mohammad El-Hajj, Osmar R. Zaiane.Parallel bifold: Largescale parallel patternmining with constraints. Distributed and Parallel Databases. 2006;20(3).

AUTHORS PROFILE

Chandaka Babi, Research Scholar in Dept of Information Technology, Gitam University, Visakhapatnam. He has completed his M.Tech from Gitam University. His Research interests includes Data Mining, Big Data Mining Data Analytics.

Dr. Mandapati Venkateswara Rao is Professor in Department of Information Technology at Gitam Institute of Technology, Gitam University, Vishakhapatnam, India. He has received M.Tech in CST and PhD in Robotics from Andhra University. His Research Interests includes Robotics, Cloud Computing and Image processing. He published several papers in International conferences and journals.

Dr.Vedula Venkateswara Rao is Professor in the Department of Computer Science Engineering at Sri Vasavi Engineering College, Tadepalligudem, India. He received Matsers Degree in Computer Science Engineering from JawaharLal Nehru Technological University Kakinada, Masters Degree In Information Technology from Punjabi University, Patiayala, India and PhD from Gitam University. His research interests include Cloud Computing and Distributed Systems, Data Mining, Big Data Analytics and Image Processing. He published several papers in International conferences and journals..

Author-4 Photo

Bhanuja Arkela B.Tech Final Year in Dept of CSC in Gitam Institute of Technology, Gitam University, Visakhapatnam India.