MapReduce Based Multilevel Association Rule Mining from Concept Hierarchical Sales Data

Dinesh J. Prajapati(✉) and Sanjay Garg

Computer Science and Engineering Department, Institute of Technology,
Nirma University, Ahmedabad, India
djprajapati.6054@gmail.com, gargsv@gmail.com

Abstract. Multilevel association rule mining is one of the important techniques of data mining to analyze the sales data. Multilevel association rules provide detailed information as compare to single level association rules. Today's era of e-commerce and e-business, various online marketing sites and social networking sites are generating tremendous amount of data in the form of sales, tweets, text mails, web usages and many more. The data generated from these sources is really too large so that it becomes tedious task to process and analyze using traditional approaches. This paper overcomes the drawback of single node computing by distributing the task to cluster of nodes. The performance of this system is analyzed using reduced minimum support threshold at different levels of concept hierarchy and by varying the database size. In this experiment, the transactional dataset is generated from big sales dataset then the distributed multilevel frequent pattern mining algorithm (DMFPM) is implemented to generate level-crossing frequent itemset using hadoop mapreduce framework. The multilevel association rules are generated from frequent itemset. The hierarchical redundant rule affects the efficiency of the system, so hierarchical redundancy is removed from it. Finally, the time efficiency of proposed algorithms is compared with existing Multilevel Frequent Pattern Mining Algorithm (MFPM).

Keywords: Distributed frequent pattern mining algorithm · Multilevel association rule · Mapreduce · Level crossing rules · Redundant rules

1 Introduction

The data mining is a technique to extract the useful knowledge and patterns from the data. One of the techniques used in data mining is called association rule mining. Association rule mining is used to discover the relationship among items. Association rule mining is used for market basket analysis where an organization is interested in identifying items that are frequently purchased together. The following terms are used in this paper.

Itemset: Let $I = \{I_1, I_2, \ldots, I_n\}$ be a set of distinct items. A set X which is subset of I is called itemset. An itemset X with k items is referred as k-itemset [1].

Support: The support is the percentage of transactions in the database D that contain both itemsets X and Y [2–4]. The equation for the support is given below.

M. Singh et al. (Eds.): ICACDS 2016, CCIS 721, pp. 624–636, 2017.
DOI: 10.1007/978-981-10-5427-3_64

$$Support = P(X \cup Y)$$

Confidence: The confidence is the percentage of transactions in the database D with itemset X also contains the itemset Y [2–4]. The equation for the confidence is given by the conditional probability is expressed in terms of itemset support.

$$Confidence = P(Y/X) = Support(X \cup Y) / Support(X)$$

Where, $Support\ (X \cup Y)$ is the number of transactions containing the itemsets X and Y both, and $Support(X)$ is the number of transactions containing the itemset X.

Association Rule: Consider a transaction database D, where each transaction T_i is a set of items, and an association rule is the relationship between those items. An association rule is expressed in the form $X \rightarrow Y$, where X and Y are the itemsets. This rule exposes the relationship between the itemset X with the itemset Y. The interestingness of the rule $X \rightarrow Y$ is measured by the support and confidence [2–4].

An itemset is frequent, if its support is greater than or equal to the user defined minimum support threshold. Association rule mining process basically consists of two steps [3–5]: (i) first, all the itemsets that satisfies the minimum support thresholds are identified and referred as frequent itemset, (ii) then, Generate strong association rules from derived frequent itemsets that satisfies minimum confidence threshold. Big data is termed for a collection of large data sets which are complex and difficult to process using traditional data processing tools. For big data, the data reading is slower from physical storage than from the recent fast network. For example, to read 1 TB of data with 10 Mbps network speed, it takes about 28 h; however, if the dataset is divided into one hundred 10 GB datasets with 10 Mbps network speed each, it takes around 17 min only [6, 7]. The complex problem can be solved using divide and conquer strategy which consists of three phases [8]: (i) When the size of problem is small enough then solve the entire problem directly; otherwise, split the original problem into two or more sub-problems. (ii) Recursively, solve the sub-problems by again applying the divide-and-conquer strategy until the sub-problem is solvable. (iii) Finally, merge the solutions of the two sub-problems to get the solution of the original problem.

Multi-level Association Rule Mining: Association rules generated from mining data at multiple levels of concept hierarchy are called multilevel association rules [9, 10]. In multiple-level association rule mining, the items are categorized by level of concept hierarchy. This concept hierarchy is used to discover the association rules at multiple concept levels.

Each node in the concept-hierarchy tree represents single item available in the itemset. The AMUL dairy dataset contains four levels of the concept hierarchy tree. Any item at level i is child of item at level $i-1$ and so on. At level one, two items Fresh Products and Frozen Products are present. Further-more, Fresh Products has two children, Milk Products and Milk. Ice-cream and Snacks are the children of Frozen Products. This hierarchy continues accordingly. While mining multiple level association rule, the dataset is encoded in a concise numeric form to identify classification level as shown in Fig. 1. Encoding the taxonomy information as a sequence of digits makes multi-level association

rule mining algorithm more efficient in terms of memory and processing. So, each node is assigned a number which represents the item id. This item id also provides encoding that gives taxonomy information about the hierarchy. The encoding is done from leftmost child, sibling and so on. For example, the item Skimmed Curd is encoded as 1122. The encoded code's first digit 1 represent Fresh Products at level 1 and second digit 1 represent Milk Products at level 2 and third digit 2 represent Curd at level 3.

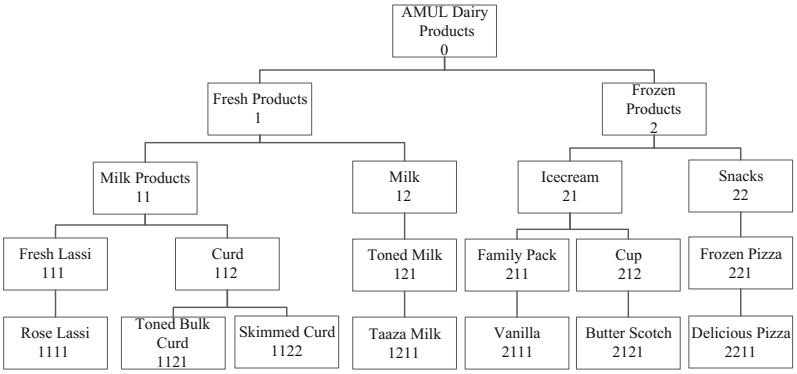

Fig. 1. Sample concept-hierarchy AMUL dairy products with taxonomy information

In brief, the contribution of this paper is summarized in three steps: (i) First of all, big sales hierarchical dataset is transformed into transactional dataset using hadoop mapreduce, (ii) The distributed multilevel frequent pattern mining algorithm is implemented to generate multilevel frequent itemsets including level crossing, (iii) Finally, Hierarchical redundant rules are eliminated to derive the interesting multilevel association rules. For the experimental purpose, multi-level frequent pattern mining algorithm and proposed algorithm are tested on sales dataset of AMUL dairy.

The remaining of this paper is organized as follows. Section 2 presents preliminaries for multilevel association rule mining from concept hierarchical sales big data in distributed environment. Related work is given in Sect. 3. Section 4 shows the proposed methodology. In Sect. 5, the performance of proposed method is evaluated on sales dataset of AMUL dairy. Finally, the conclusions and future scope is drawn in Sect. 6.

2 Preliminaries

This section covers the complete set of definitions, terminologies and assumptions used in this paper.

2.1 Data Pre-processing

Data pre-processing is the process of solving irrelevant or missing information and change the basic structure of data, collected from various sources [11]. In real world, the dataset may contain data in the format that cannot be used for further processing; such dataset can be converted into desired form using preprocessing.

2.2 Hadoop

Hadoop is an open source framework which implements the mapreduce programming model in the distributed environment [12]. In the hadoop framework, computer machine or node is classified as master node and slave nodes. The master node supervises data storage on machine and running parallel computations on that data. The slave nodes are the machines which perform the work of storing the data and running the computations.

2.2.1 Hadoop Distributed File System (HDFS)

The Hadoop runtime system is coupled with HDFS which provides parallelism and concurrency to achieve system reliability. HDFS is designed for storing huge files with streaming data access patterns, running on clusters of commodity hardware [12].

2.2.2 Mapreduce

Mapreduce is basically software introduced by Google to support distributed computing on big datasets using the cluster of nodes. The mapreduce framework consists of two functions [13]: Mapper and Reducer. The mapper and reducer function is basically written by the user. The data structure of map and reduce function is defined by <*key, value*> pairs. The mapper function takes an input as <*key, value*> pair and produces a set of intermediate result as <*key, value*> pairs. The reducer function receives an intermediate key generated and a set of values for that key. The reducer merges these values of the key to generate smaller set of values. The output of reducer is value zero or one typically.

2.3 Performance Analysis

The proposed approach can be evaluated on the basis of execution time of the algorithm, scalability & flexibility of data, and data heterogeneity to analyze the performance of the system [14].

2.3.1 Execution Time

The execution time is the time required to perform data mining or analysis task. The efficiency and time complexity of the algorithm is computed based on time required for getting the desired output quickly and efficiently.

2.3.2 Scalability and Flexibility

The performance of the system should not scale down even if data is scaled from thousands to billions of records. The system should be more flexible enough to handle heavy workload fast.

2.3.3 Data Heterogeneity

The data is coming from various sources due to the ease of internet and social media. The basic types of dataset may be of structured, semi structured and unstructured. Ideally, the system should accept heterogeneous data as an input then preprocess the data and finally, produce the desired output within stipulated time.

3 Related Work

Han and Fu [10] present various interestingness measures to find more interesting rules including level crossing rules. In this paper multiple level association rules discovers the interesting and strong rules from the large database. Authors also suggested modified methods for mining single level association rules to multiple level association rules which creates interesting issues for the further work. Thakur et al. [15] proposed a top-down approach for mining multilevel level-crossing association rules from the large transaction databases by using extension of existing approaches. In this paper, authors have used the concept of reducing support as well as filtered the transaction table, T for each levels of concept hierarchy. After generating a new filtered transaction tables at one concept level, similar process will be carried out for remaining level. This approach improves the processing time and generates less candidate itemsets. Wan et al. [16] proposed a novel approach to improve the efficiency, integrality and accuracy by analyzing multiple level association rules from primitive concept level of hierarchy. In this paper, the proposed method considers the dynamic concept hierarchies to generate multilevel association rules from customized point of view. The paper also mentioned various issues for the calculation of rule support and multilevel association rules at specific level.

The authors in [17, 18], proposed a method to remove the hierarchical redundancy using frequent closed itemsets. In this paper, hierarchical redundancy is removed to reduce the basic size of the association rules which improves the quality and usefulness of rule without losing any information. Author also suggests that this approach can be apply to the approximate basis rule to remove the redundancy. Hong et al. [19] proposed an incremental multilevel association rule mining algorithm based on the pre-large concept hierarchy with taxonomy information. The large frequent itemsets plays an important role to reduce database scan. Due to repeatedly scanning of the database, the efficiency of algorithm decreases. The author proposed algorithm to reduce the mining cost. Gautam and Pardasani [20] proposed boolean matrix based approach to discover frequent itemsets. The proposed approach scans the transaction database only one time and does not produce itemsets, but adopts the boolean vector relational calculus to discover frequent itemset. Boolean matrix based approach stores all transaction data in bits, so it needs less memory space. Ramana et al. [21] evaluated and compared traditional multilevel association rule mining algorithms like ML_T2L1, ML_T1LA, ML_TML1 and ML_T2LA. Algorithm ML_T2L1 finds multilevel large frequent itemsets from transactional database. ML_T1LA algorithm uses only single encoded transaction table. ML_TML1 algorithm generates multiple encoded transaction tables. ML_T2LA algorithm uses two encoded transaction tables and integrates the optimization techniques to find rules. Prakash et al. [22] proposed new approach to mine both the frequent and in-frequent interesting association rules without generating redundant rules. The proposed approach discovers the association rules that are complete according to propositional logic from a given dataset. The limitation of this approach is that if an unclassified dataset is used than classification must be performed before mining the rules.

Gautam and Pardasani [23] proposed partition and boolean based method to find frequent itemsets at each concept levels to reduce the number of database scans, I/O cost and CPU over-head. A top-down approach is used for efficient mining of multi-level rules. The algorithm proposed in this paper, uses boolean AND operator to reduce the time by removing unnecessary candidate itemset. The partitioning method is used to overcome the limitation of memory requirement. Gautam and Shukla [24] proposed a method for mining multilevel association rules using reduced minimum support threshold at each level. The authors have used the pincer search algorithm to mine multilevel frequent itemsets in a given transactional database. The algorithm presented in this paper, reduces both the number of database scan and candidates itemset; thus the time efficiency is improved. Karim et al. [25] proposed a distributed syatem for mining the transactional datasets using an improved mapreduce framework. In this paper, authors implemented "Associated-Correlated-Independent" algorithm to find the complete set of customer's purchase patterns along with the correlated, associated, associated-correlated, and independent purchase patterns. Butincu and Craus [26] present improved version of the frequent itemset mining algorithm as well as its generalized version. The authors introduced optimized formulas for generating valid candidates by reducing number of invalid candidates. By using the computations of previous steps by other processed nodes, it avoids generating redundant candidates. Authors also suggested to run the same algorithm in parallel or distributed system. Chandanan and Shukla [27] proposed an algorithm to remove hierarchical duplicate rules in multi-level using upper level closed frequent itemset and generator. The algorithm proposed in this paper, reduces the size of the rules to achieve good quality and improve the usefulness of rule without information loss. The basic goal behind this approach is to improve the time efficiency by removing the hierarchically redundant rules. Pumjun and Kreesuradej [28] proposed MLUpCS algorithm to mine multilevel association rules in dynamic databases under the different support threshold without rescanning of a whole dataset. This algorithm is extension the MLUp algorithm which mines multilevel association rules using the same minimum support threshold.

However, none of the above mentioned work deals with the problem of transforming sales data into transactional data and multilevel association rule mining including level crossing using mapreduce. Hence, mapreduce based data transformation is the initial part of this work then distributed multilevel frequent pattern mining algorithm is implemented to generate level-crossing frequent itemsets. Existing MFPM [15] algorithm generates large candidate set and its execution time is too high while dealing with big data. The proposed algorithm improves the drawback of existing multilevel frequent pattern mining algorithm and also improves the execution time of system by generating small candidate itemset.

4 Proposed Methodology

The overall architecture of the proposed methodology is shown in Fig. 2. Big hierarchical sales dataset of AMUL dairy is given as input to pre-processing unit to transform it into transactional dataset using hadoop mapreduce. These generated transactional dataset is given as input to the distributed frequent pattern mining algorithm for varying

minimum support threshold which generates frequent k-itemset. Then, Multilevel association rules including level crossing rules are generated from it and finally hierarchical redundancy has to be removed to improve the performance of system. For this experiment, the hierarchical sales database of AMUL dairy having total size of 5 GB is used. The dataset contains more than 1500 different dairy products.

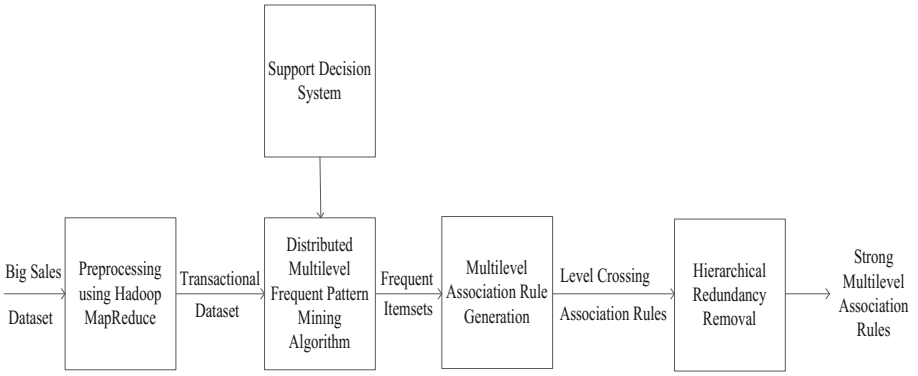

Fig. 2. Proposed methodology

4.1 Distributed Multilevel Frequent Pattern Mining (DMFPM) Algorithm

The existing multilevel frequent pattern mining algorithm generates large candidate itemset and execution time is also higher while dealing with big data. These drawbacks can be rectified by distributed multilevel frequent pattern mining algorithm proposed in this paper. Distributed multilevel frequent pattern mining algorithm is implemented to find frequent itemsets from the actual transactional dataset. Once the actual transactional dataset is stored in HDFS, the entire dataset is split into smaller segments. Each segment is transformed to the data nodes. The primary advantage of this approach is that, it exchanges the count values between each node rather than exchanging the data. The map function is executed on each data segment and it produces <*key, value*> pairs including level-crossing for each transaction of dataset. The mapreduce framework makes group of all <*key, value*> pairs having same items and executes the Reducer function by passing the list of values for candidate itemsets. The map function generates local candidate itemsets, and then the Reduce function gets global counts by adding individual local counts. For the overall computation, multiple iterations of mapreduce functions are necessary.

The distributed frequent pattern mining algorithm DMFPM shown in Fig. 3, uses notation *C[l, k]* as a set of candidate *k*-itemset at level *l* and *L[l, k]* as a set of frequent *k*-itemset at level *l*. The transactional data is given as an input to the mapper, line by line. Each line is split into itemset which is further split into items. Mapper generates

the output <*key, value*> pair, where key is the item set and value is 1. Value here indicates the local frequency of the itemset. The reduce task combines the output of the mapper and generates frequent itemset at that level. For each level, the mapreduce function produces a frequent itemset including level-crossing at that specific level. The iteration continues until no further frequent itemsets are found for that level. Frequent itemsets are calculated based on different values of minimum support threshold at each level.

Input: Database in HDFS containing encoded concept hierarchy information (D),
Maximum level of concept hierarchy (*Max_level*),
Minimum Support Threshold for each level l (*Min_sup [l]*).
Output: $L\ [\ l\ ]$, Level-crossing frequent itemsets for each level l.
Method:
For each level l in concept hierarchy do
$L[\ l,1\ ]$ = find frequent1-itemsets from (D).
For each frequent k-itemset in level l do
$C[\ l, k\ ] = L[\ l, k\text{-}1\ ] \bowtie L[\ l, k\text{-}1\ ]$.
If ($l > 1$) then
For j=1 to $l\text{-}1$ do
$C[\ l, k\ ] += L[\ j, k\text{-}1\ ] \bowtie L[\ l, k\text{-}1\ ]$.
$CT[\ l, k\ ]$ = Apply Map function on $C[\ l, k\ ]$.
$L[\ l, k\ ]$ = Apply Reduce function on $CT[\ l, k\ ]$.
$L\ [\ l\ ] = L\ [\ l\ ]\ U_k\ L[\ l, k\ ]$.

Map Function:
Input: Transaction T_i
Output: < *candidate itemset, value*>
Method:
For each transaction $T_i \in D$ do
For each itemset S_i in Candidate Itemset do
For each item $I_i \in S_i$ do
n = String_length (I_i).
If ($Sub\_string(I_i, n) \notin T_i$) then
Terminate the current itemset S_i.
Generate the output < S_i, 1 > as < *key, value* > pair.
Reduce Function:
Input: < *candidate itemset, list* >
Output: < *frequent itemset, support_count* >
Method:
count = 0.
For each number in *list* do
count += *number*.
If (*count* >= *Min_sup*) then
Generate the output < *frequent itemset, count* > as < *key, value* > pair.

Fig. 3. The DMFPM algorithm

4.2 Multilevel Association Rule Generation

The output of distributed multilevel frequent mining algorithm is frequent itemsets at each level of concept hierarchy. These generated multilevel frequent itemset is given as input to the multilevel association rule generator module to generate meaningful multilevel association rules which satisfies minimum confidence threshold. Multilevel association rules can be generated as follows [3, 29].

- For each level of concept hierarchy,
  - For each level-crossing frequent k-itemset, f, generate all non-empty subsets of f.
  - For every non-empty subset s of f, generates the multilevel association rule as $s \rightarrow (f - s)$ such that $(Support\ (f)/Support\ (s)) \geq min\_conf$, where min_conf is the minimum confidence threshold at that level.

Since, the rules are generated from multilevel frequent itemsets; each rule automatically satisfies minimum support threshold.

4.3 Eliminating Hierarchical Redundant Rules

The hierarchical redundant rules are generated due to ancestor relationship among the items. The processing of such redundant rules degrades performance of the system. Hence, hierarchical redundant rules are eliminated to improve the quality and usefulness of the rules without loss of information. An association rule R_1 is an ancestor of another association rule R_2 if rule R_1 can be obtained by replacing the items in the rule R_2 by their ancestors in a concept hierarchy [3]. In such case, rule R_2 is not interesting since it does not provide new information and is less general than first rule R_1. Such rule is redundant and need to be eliminated.

5 Experimental Setup and Results

For the experimental purpose, a cluster of four desktop machines consisting of i5 processor with 4 GB DDR-3 RAM are used. Ubuntu 12.04 LTS operating system is installed in all the four nodes. Usually JVM is not a part of Ubuntu 12.04, so, JVM is also installed in all the nodes. Multi-node cluster is configured in three computers and single-node cluster is configured in a single computer using apache hadoop packages. The distributed multilevel frequent pattern mining algorithm is tested on both multi-node as well as single-node cluster and compared with existing algorithm.

5.1 Generation of Multilevel Frequent Pattern

After transforming transactional dataset into actual transactional dataset, actual transaction file is given as input to the proposed algorithm to find the frequent itemsets without level-crossing and including level-crossing.

5.1.1 Multilevel Frequent Pattern Mining Without Level Crossing

When a uniform minimum support threshold is used at each level of concept hierarchy, the search procedure is simplified but some of the interesting multilevel association rules may be missed. So, minimum support threshold is adjusted such that it reduces from higher level to lower level of the concept hierarchy. The MFPM and DMFPM algorithms are applied on 5 GB AMUL dataset using single node. For reducing minimum support threshold level–wise, the execution time for the MFPM and DMFPM algorithms is shown in Table 1. The level wise minimum support threshold 4-3-2-1 indicates *min_sup* of 4% for level 1, 3% for level 2, 2% for level 3 and 1% for level 4, respectively. It can be observed that the execution time of the proposed algorithms is significantly lower as compared to MFPM algorithm [15].

Table 1. Level-wise minimum support threshold vs execution time

Level-wise min. support threshold (%)	Execution time (in seconds)	
	The MFPM algorithm [15]	The DMFPM algorithm
4-3-2-1	36789	6800
5-4-3-2	31256	4800
6-5-4-3	24567	3900
7-6-5-4	16788	1987
8-7-6-5	13567	1178

5.1.2 Multilevel Frequent Pattern Mining Including Level Crossing

For finding multilevel frequent itemset including level-crossing, the minimum support threshold is considered similar for all the levels of concept hierarchy. The MFPM algorithm is applied on single node and DMFPM algorithm is applied on single, two as well as on three node cluster. The minimum support threshold is considered as 1%. The result of MFPM and DMFPM algorithms on AMUL datasets for the varying database size 256 MB, 512 MB, 1 GB, 2 GB and 5 GB is shown in Table 2. For a data set of size 5 GB that was distributed on single node, the execution time for the MFPM and DMFPM algorithms are 68000 s and 4800 s respectively. The experiment shows that the execution time of proposed algorithms is less as compared to the MFPM algorithm.

Table 2. Dataset size vs execution time

Dataset size (in MB)	Execution time (in seconds)			
	The MFPM algorithm [15]	The DMFPM algorithm (single node cluster)	The DMFPM algorithm (two node cluster)	The DMFPM algorithm (three node cluster)
256	3189	280	221	191
512	5000	490	380	258
1024	14980	896	696	405
2048	24000	1940	1640	1040
5120	68000	4800	4154	2300

The proposed algorithms provide much better performance as compared to MFPM when the size of the dataset is large. Furthermore, the efficiency of proposed algorithm is improved using cluster of nodes.

5.2 Hierarchical Interesting Association Rule Generation

Once the level-crossing association rules are generated for each level then, hierarchical redundant rules are eliminated from it to improve the efficiency. For this experiment, the number of hierarchical redundant association rules is calculated for minimum confidence threshold 40%, 50%, 60%, 70%, 80% and 90%, and minimum support threshold 1%, 2%, 3%, 4% and 5%, as shown in Fig. 4. It is observed from the experimental result that the number of hierarchically redundant rules generated is less when minimum confidence threshold and minimum support threshold are more than 70% and 3%, respectively.

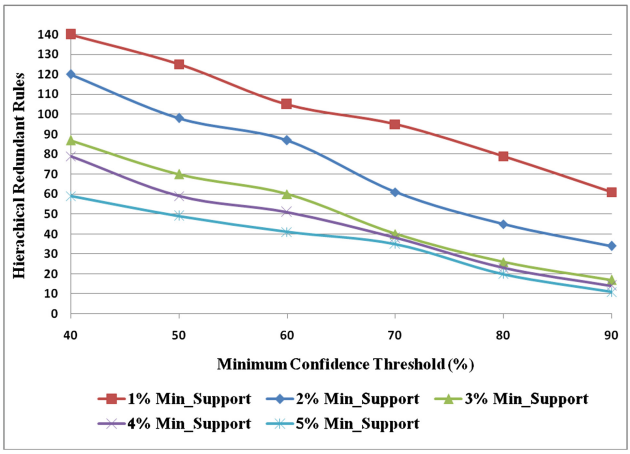

Fig. 4. Minimum confidence threshold vs hierarchical redundant rules

6 Conclusions and Future Scope

Traditional multilevel association rule mining algorithms have limitations of processing speed while analyzing the big data. HDFS and mapreduce play an important role in pre-processing, handling and analysis of such data. In this paper, hadoop based distributed approach is presented which process data by partitioning into cluster of nodes. The primary goal of this work is to reduce inter-node message passing in the cluster. In this paper, the proposed algorithm is used to mine multilevel association rules at same level and different levels of concept hierarchy. The proposed algorithm generates less number of candidate itemset and uses less message passing. Hence, the execution time of the proposed algorithms is comparatively less. The experimental results show that

the distributed frequent pattern mining algorithm scale linearly with increasing database size. From the experimental results, it is observed that in order to reduce execution time, the number of node must increase accordingly. Furthermore, for the higher value of minimum confidence threshold and minimum support threshold, number of hierarchically redundant rules is less. The proposed algorithm is more flexible, scalable and efficient distributed multilevel frequent pattern mining algorithms for mining big data.

The time efficiency of the proposed algorithms can be yet improved by reducing the number of database scans for each level of concept hierarchy.

Acknowledgements. The authors take this opportunity to thank all the researchers from the domain of big data analysis for their immense knowledge and kind support throughout the work. Also would like to thank our institute for their resources and constant inspiration. Special thanks to the authority of AMUL dairy located at Anand, Gujarat, India for providing hierarchical big sales database. At last heartiest thanks to our family and friends for encouraging us to make this a success.

References

1. Srikumar, K., Bhasker, B.: Metamorphosis: mining maximal frequent sets in dense domains. Int. J. Artif. Intell. Tools **14**(3), 491–506 (2005)
2. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. In: International Conference of ACM-SIGMOD on Management of Data, pp. 207–216 (1993)
3. Han, J., Kamber, M.: Data Mining Concepts & Techniques. Morgan Kaufmann Publishers, San Francisco (2004)
4. Olsan, D.L., Delen, D.: Advanced Data Mining Techniques. Springer, Heidelberg (2008)
5. Tseng, F.S.C., Chen, P.Y.: Parallel association rule mining by data de-clustering to support grid computing. Proc. PACIS **89**, 1071–1084 (2005)
6. Woo, J., Basopia, S., Kim, S.H.: Market basket analysis algorithm with NoSQL DB HBase and Hadoop. In: 3rd International Conference Emerging Databases (EDB2011), Korea, pp. 56–62 (2011)
7. Woo, J., Basopia, S., Kim, S.H.: Market basket analysis algorithm with map/reduce of cloud computing. In: Proceedings of the International Conference Parallel and Distributed Processing Techniques and Applications, USA (2011)
8. Tseng, F.S.C., Kuo, Y.H., Huang, Y.M.: Toward boosting distributed association rule mining by data de-clustering. Inf. Sci. **180**(22), 4263–4289 (2010). Elsevier
9. Angryk, R.A., Petry, F.E.: Mining multi-level associations with fuzzy hierarchies. In: The 14th IEEE International Conference on Fuzzy System, pp. 785–790 (2005)
10. Han, J., Fu, Y.: Mining multiple-level association rules in large databases. IEEE Trans. Knowl. Data Eng. **11**(5), 1–8 (1999)
11. Chang, F., Dean, J., Ghemawat, S., Hsieh, W.C., Wallach, D.A., Burrows, M., Gruber, R.E.: Bigtable: a distributed storage system for structured data. ACM Trans. Comput. Syst. (TOCS) **26**(2), 1–14 (2008)
12. Apache Hadoop. http://hadoop.apache.org/
13. Yeung, J.H.C., Tsang, C.C., Tsoi, K.H., Kwan, B., Cheung, C., Chan, A.P.C., Leong, P.H. W.: Map-reduce as a programming model for custom computing machines. In: 16th IEEE Symposium on Field-Programmable Custom Computing Machines FCCM 2008 (2008)

14. Jagdale, A.R., Sonawane, K.V., Khan, S.S.: Data mining and data pre-processing for big data. Int. J. Sci. Eng. Res. **5**(7), 1156–1161 (2014)
15. Thakur, R.S., Jain, R.C., Pardasani, K.R.: Mining level-crossing association rules from large databases. J. Comput. Sci. **2**(1), 76–81 (2006)
16. Wan, Y., Liang, Y., Ding, L.: Mining multilevel association rules with dynamic concept hierarchy. In: Proceedings of the 7th International Conference on Machine Learning and Cybernetics pp. 287–292. IEEE (2008)
17. Shaw, G., Xu, Y., Geva, S.: Eliminating redundant association rules in multilevel datasets. In: 4th International Conference on Data Mining, Las Vegas, USA, pp. 14–17 (2008)
18. Xu, Y., Shaw, G., Li, Y.: Concise representations for association rules in multilevel datasets. J. Syst. Sci. Syst. Eng. **18**, 53–70 (2009). Springer
19. Hong, T., Huang, T., Chang, C.: Mining multiple-level association rules based on pre-large concepts. In: Data Mining and Knowledge Discovery in Real Life Applications Austria, pp. 187–200 (2009)
20. Gautam, P., Pardasani, K.R.: A fast algorithm for mining multilevel association rule based on Boolean matrix. Int. J. Comput. Sci. Eng. **2**(3), 746–752 (2010)
21. Ramana, V.V., Rathnamma, M.V., Reddy, A.R.M.: Methods for mining cross level association rule in taxonomy data structures. Int. J. Comput. Appl. **7**(3), 28–35 (2010)
22. Prakash, S., Vijayakumar, M., Parvathi, R.M.S.: A novel method of mining association rule with multilevel concept hierarchy. Int. J. Comput. Appl. (IJCA), 26–29 (2011)
23. Gautam, P., Pardasani, K.R.: Efficient method for multiple-level association rules in large databases. J. Emerg. Trends Comput. Inf. Sci. **2**(12), 722–732 (2011)
24. Gautam, P., Shukla, R.: An efficient algorithm for mining multilevel association rule based on Pincer search. Int. J. Comput. Sci. Issues (IJCSI) **9**(4), 235–241 (2012)
25. Karim, M.R., Ahmed, C.F., Jeong, B., Choi, H.: An efficient distributed programming model for mining useful patterns in big datasets. IETE Tech. Rev. **30**(1), 53–63 (2013)
26. Butincu, C.N., Craus, M.: An improved version of the frequent itemset mining algorithm. In: Proceedings of the 14th IEEE International Conference Networking in Education and Research, Craiova, pp. 184–189 (2015)
27. Chandanan, A.K., Shukla, M.K.: Removal of duplicate rules for association rule mining from multilevel dataset. Procedia Comput. Sci. **45**, 143–149 (2015). Elsevier
28. Pumjun, N., Kreesuradej, W.: Maintenance of multi-level association rules discovery in dynamic database under a change of support threshold. In: 12th IEEE International Confernce on Fuzzy Systems and Knowledge Discovery (FSKD), pp. 618–623 (2015)
29. Ban, T., Eto, M., Guo, S., Inoue, D., Nakao, K., Huang, R.: A study on association rule mining of darknet big data. In: Proceedings of the IEEE International Joint Conference on Neural Network (IJCNN), pp. 1–7 (2015)