Implementation of Enhanced Apriori Algorithm with Map Reduce for Optimizing Big Data

Sunil Kumar Khatri[1] and **Diksha Deo**[2]

Submitted in June 2014; Accepted in May, 2015

Abstract — Nowadays as a result of speedy increase in data technology. Massive scale processing may be a major purpose of advanced technology. To handle with this advance progress in information assortment and storage technologies, designing, and implementation massive scale algorithms for data processing is gaining quality and big interest. In data processing domain, association rule classification and learning may be a common and well researched methodology for locating fascinating relations between variables in massive databases. Apriori is that the key algorithmic rule to get the frequent item sets. Analyzing frequent item sets may be a crucial step to find rules and association between them. This stands as a primary foundation to monitored learning, which incorporates classifier and have extraction strategies. Enforcing this algorithmic rule is crucial to infer the behavior of structured information. In scientific domain, most of the structured information in are voluminous. Process such reasonably Brobdingnagian information needs special and dedicated computing machines. fitting such associate degree infrastructure is troublesome and dearly-won. Association rule mining needs massive computation and I/O traffic capability.

This paper majorly focuses on making association rules and Map/Reduce style and implementation of Apriori for structured information. Optimize Apriori algorithmic rule to scale back communication value. This paper aims to extract frequent patterns among set of things within the dealing info or different repositories. Apriori algorithmic rule contains a nice influence for locating frequent item sets victimization candidate generation. Apache Hadoop Map cut back is employed to create the cluster. It operating relies on Map cut back programming modal. it's accustomed improve the potency and process of enormous scale information on high performance cluster. It additionally processes Brobdingnagian information sets in parallel on massive cluster of pc nodes. It provides reliable, ascendable, distributed computing.

Index Terms— Big Data, Map Reduce, Apriori Algorithm, Optimization.

1.0 INTRODUCTION

Big data is a big thing and growing rapidly and a significant interest in new analytics used in big data. To describe big data we can say that it comprises of transactions, interactions and observations. Big data introduces large volumes of unstructured data. This data changes is highly dynamic and therefore needs to be ingested quickly for analysis.

Big knowledge could be a massive factor and growing speedily and significant interest in new analytics employed in big knowledge. To explain massive knowledge we are able to say that it contains of transactions, interactions and observations. Massive knowledge introduces massive volumes of unstructured knowledge. This knowledge changes is extremely dynamic and so must be eaten quickly for analysis.

In several applications of the important world, generated knowledge is of nice concern to the neutral because it delivers purposeful data that assists in creating prophetic analysis. This data helps in modifying sure call parameters of the applying that changes the outcome of a business method.

The volume of knowledge, conjointly referred to as data-sets, generated by the application is terribly massive. So, there is a requirement of process massive data-sets expeditiously. The knowledge-set collected is also from heterogeneous sources and will be structured or unstructured data. Process such knowledge generates helpful patterns from that data will be extracted. The only approach is to use this model and insert headings and text into it as acceptable.

There are totally different layers of massive knowledge hierarchy:

1. Variety of data
   - Structured
   - semi-structured
   - unstructured
   - complex
2. Sort of data: XML datasets.
3. Rate of data: Real time, close to real time, each minute, every hour, daily.
4. Volume of data: computer memory unit, Petabytes, Terabyte.
5. Sorts of analysis: Classification analysis, pattern recognition, regression, text-mining, clustering, anomaly detection.

The need to implement and derive sensible optimization technique to vary the manner the info is ruled. Manufactures are started synchronizing and analyzing the massive datasets for future call and to grow their business and production productively. Optimization technique has many capabilities that build it a perfect alternative for knowledge analysis in such state of affairs. Firstly, this method is intended for analyzing and drawing insights for extremely advanced system with immense knowledge volumes, multiples constraints and factors to be proclaimed for. Secondly, market has totally different

[1, 2]*Amity Institute of Information Technology, Amity University Uttar Pradesh, India*
E-mail: [1]skkhatri@amity.edu and [2]dikshadeo@gmail.com

range of enterprise objective related to it like value reduction, demand fulfillment etc.

A. Alternative of algorithmic program:

One among the foremost necessary things is to settle on acceptable algorithm for optimizing massive knowledge.

B. Steps to method massive data:

1. Classic ETL process.
2. Knowledge discovery and Investigate analysis: interactive knowledge exploration
3. Massive knowledge refinery: Store, mixture and remodel multi structured knowledge to correct worth.
4. Share refined knowledge and runtime modal.
5. Retain runtime modals and historical knowledge for current refinement and analysis.
6. Metallic element (Business intelligence) and analytics: retain historical knowledge to unlock extra worth. Like dashboard and good image analytics.

2.0 LITERATURE REVIEW

Distributed information Mining in Peer-to-Peer Networks (P2P) [1] offers associate degree summary of distributed data-mining applications and algorithms for peer-to-peer surroundings. It describes both exact and approximate distributed data-mining algorithms that work in a decentralized manner. It illustrates these approaches for the matter of computing and observation clusters within the information residing at the completely different nodes of a peer-to-peer network. This paper focuses on associate degree rising branch of distributed information mining known as peer-to-peer information mining. It additionally offers a sample of actual and approximate P2P algorithms for bunch in such distributed environments.

Web Service-based approach for information mining in distributed environments [2] presents associate degree approach to develop a knowledge mining system in distributed environments. This paper presents a net service-based approach to solve these issues. The system is engineered victimization this approach offers a uniform presentation and storage mechanism, platform autonomous interface, associate degreed an dynamic protractile design. The planned approach during this paper permits users to classify new incoming information by choosing one amongst the antecedently learnt models.

Architecture for data processing in distributed environments [3] describes system design for climbable and transportable distributed data processing applications. This approach presents a document image known as imp for accessing and looking out for digital documents in fashionable distributed info systems. The paper describes a corpus linguistic analysis of giant text corpora primarily based on collocations with the aim of extracting linguistics relations from unstructured text.

Distributed information Mining of giant Classifier Ensembles [4] presents a new classifier combination strategy that scales up expeditiously and achieves each high prognostic accuracy and trait of issues with high ramification. It accelerates a world model by discovering from the averages of the native classifiers output. The effective combination of enormous variety of classifiers is achieved this fashion. Multi Agent-Based Distributed information Mining [5] is the integration of multi-agent system and distributed data processing (MADM), additionally referred to as multi-agent primarily based distributed data processing. The angle here is in terms of implication, system's read, existing systems, and analysis tendencies. This paper presents an outline of MADM systems that are conspicuously in use. It additionally defines the common elements between systems and offers an outline of their methods and design.

Preserving Privacy and sharing the information in Distributed atmosphere victimization cryptographically Technique on rattled information [6] proposes a framework that enables systematic transformation of original information victimization randomized information perturbation technique. The changed information is then submitted to the system through cryptographically approach. This method is applicable in distributed environments wherever every information owner has his own information and needs to share this with the opposite information homeowners. At an equivalent time, this information owner needs to preserve the privacy of sensitive information within the records.

Distributed anonymous information perturbation technique for privacy-preserving information mining [7] discusses a light-weight anonymous information perturbation technique for economical privacy conserving in distributed data processing. 2 protocols are planned to deal with these constraints and to safeguard information statistics and also the organization method against collusion attacks. Associate degree algorithmic rule for Frequent Pattern Mining supported Apriori [8] proposes 3 completely different frequent pattern mining approaches (Record filter, Intersection and also the planned Algorithm) supported classical Apriori algorithmic rule. This paper performs a comparative study of all 3 approaches on a data-set of 2000 transactions. This paper surveys the list of existing association rule mining techniques and compares the algorithms with their changed approach.

Using Apriori-like algorithms for Spatio-Temporal Pattern Queries [9] presents a approach to construct Apriori-like algorithms for mining Spatio-temporal patterns. This paper addresses issues of the completely different sorts of examination functions that will be used to mine frequent patterns. Map-Reduce for Machine Learning on Multi core [10] discusses ways to develop a broadly applicable parallel programming paradigm that is applicable to different learning algorithms. By taking advantage of the summation kind during a map-reduce framework, this paper tries to pose a large vary of machine learning algorithms and reach a major speed-up on a twin processor cores. Victimization Spot Instances for Map cut back Work flows [11] describes new techniques to improve the runtime of Map cut back jobs. This paper presents Spot Instances (SI) as a means of attaining performance gains at low monetary cost.

3.0 PROPOSED ENHANCED APRIORI ALGORITHM

Apriori algorithm finds all frequent item sets by scanning the database time after time. This algorithm consumes a lot of time and memory space so to present parallelization in Apriori algorithm, an bettered Apriori algorithm is proposed which is shown below:-

1. In opening, parallel scan initial split the group action information horizontally into‘s’ node subsets and distribute it to‘t’ nodes supersets.
2. The various‘s’ nodes are then processed more. Main concern is to separate into major chunks of information that's reticulated inside.
3. Then once the method is completed, every node scans its own information sets then generates set of Candidate item set Kp.
4. Then the support count of every Candidate item set is about to one. This Candidate item set Kp is split into r partitions and sent to ‘r’ nodes with their support variety count. ’r’ nodes severally conglomerate the support variety count of an equivalent item set to provide the ultimate sensible support, and influence the frequent item set Fp within the partition once examination with the minimum support min_sup.
5. Once the process of every candidate item set, Kp is split more to calculate the frequent itemset exploitation pruning.
6. Finally merge the output of ‘r’ nodes to come up with set of world frequent item set K. This impermanent algorithmic rule is employed to significantly scale back the time as during this algorithmic rule obtaining frequent item sets by traversing the transactional information only once. The performance degradation with Sector filing system is perhaps thanks to I/O overhead wherever there's an excellent input and output transactions of file transfer between the native and distributed filing system. Another potential vital issue behind this could be JNI (Java Native Interface) overhead because it faces issue whereas researching JNI Layer to access Sector. Once merging we tend to not solely save time interval however additionally scales back the quantity of processors interactions throughout the execution of program.

There are some of the known methods to improve the efficiency of Apriori Algorithm using parallel dynamic Itemset count: Add new candidate item sets only when all of their subsets are estimated to be frequent. Otherwise, drop that item set or dump that data. [12]

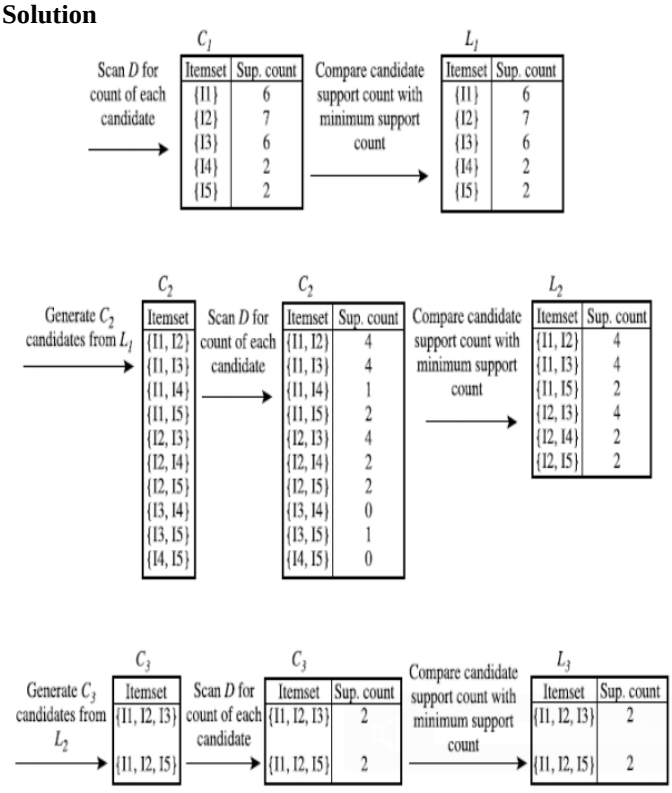
4.0 ANALYSIS OF EXISTING APRIORI ALGORITHM AND PROPOSED APRIORI ALGORITHM

In this part, we will compare time efficiency in finding frequent item set by normal Apriori algorithm and by method proposed in this paper. Now, consider the following example and calculate time to generate frequent item sets by using basic Apriori algorithm.

Problem: This problem justifies the legacy statement, how actually it works and process for a valid output. [10]

TID	List of item_IDs
T100	I1,I2,I5
T200	I2,I4
T300	I2,I3
T400	I1,I2,I4
T500	I1,I3
T600	I2,I3
T700	I1,I3
T800	
I1,I2,I3,I5	

Solution



As discussed, the existing Apriori algorithm has many loopholes, they are as follows:-

1. Historical Significance.
2. Suffers from a number of insufficiency and trade-offs.
3. Uses a minimum number of support thresholds.

According to the proposed methodology, the comparisons are based on using the native Java application. Table1. Indicates the results that will show how many records of n transaction can generate how many number of association rules.

Table 1: Proposed Apriori algorithm: No. of rules generation

No. of Records	Support	Confidence	Minimum support threshold	Generation time (seconds)	Number of frequent sets	No. of rules
1800	20	50	20% (360 records)	0.01	107	512
1750	20	50	20% (350 records)	0.01	107	425
51	20	50	20% (10.2 records)	0.00	55	210

```
Number of records = 1800
Number of columns = 10
---------------------------------------
INPUT SUPPORT THRESHOLD:
Support threshold (%) = 20.0
Minimum support (# records) = 360.0
---------------------------------------
INPUT CONFIDENCE THRESHOLDS:
Confidence threshold (%) = 50.0
---------------------------------------
SORT INPUT DATA:
---------------------------------------
APRIORI-GEN:
Minimum support threshold = 20.0% (360.0 records)
Generation time = 0.01 seconds (0.0 mins)
Number of frequent sets = 107.
---------------------------------------
```

After tested on various performance analysis portals, we came to a conclusion that the existing algorithm is not as efficient and reliable enough to use. To use it for big data optimization, we have to make sure that all parameters are covered properly and then we can implement further.

Table 2: Performance evaluation between existing and proposed algorithm

Itemset	Efficiency/ Performance Improvement	Existing Apriori Algorithm	Proposed Apriori Algorithm
800	20%	0.08 s	0.02s
1200	34%	0.20s	0.08s
1800	42%	0.40s	0.10s
2000	48%	0.50s	0.30s

5.0 IMPLEMENTATION OF PROPOSED METHOD

5.1 Pseudocode : Apriori Algorithm

The aprioriGen() accepts the set of all (k-1) large item set as parameters and returns a superset of the k large itemset as the candidate set. The outermost for Loop keep repeating k to further generate the candidate set for all levels of large itemset. Observing that the (k-1) itemset statement is detected as global huge set, thus we are not able to make a meaningful disseminated form of this function, and for sure, we can replicate the (k-1) large item sets among all the sites, but this is little obvious and thus makes no significance.

Step 1: Joining- Fk is generated by joining P(k-1) with itself.

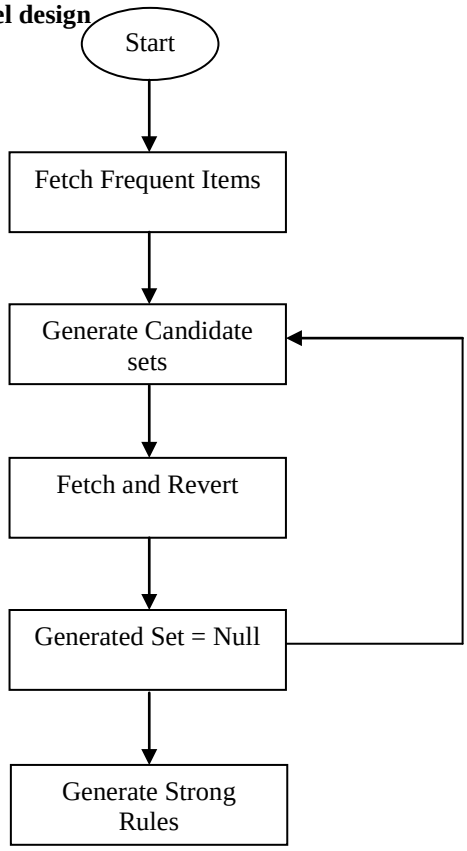
Step 2: Pruning- Any (k-1) - itemset that is not frequent cannot be a subset of a frequent k-itemset.

Step3: Generating Rules- frequent itemset then generates strong association rules.

Algorithm:

1. *pk: Candidate thing set P of size k.*
2. *sk: Frequent thing set S of size k.*
3. *f1 = {processed continuous items};*
4. *for(k= 1;sk!=Ø; k++) do Begin*
5. *ck+1= hopefuls created from Fk;*
6. *for every transaction t in database does augment the include of all applicants Ck+1 that are held in transaction t.*
7. *then, Fk+1 = applicants in Ck+1 with min_support limit esteem.*
8. *end;*
9. *return F*

5.2 High level design



6.0 IMPLEMENTATION WITH MAP REDUCE

The Map Reduce is a distributed Programming framework projected for massive cluster of systems arrangements that will operate in parallel on a really Brobdingnagian dataset. The task huntsman is to blame for managing the Map Reduce method. The tasks separated by the most application are foremost processed by the map tasks in a wholly parallel manner. The Map cut back framework kinds the maps output, that are then use as associate degree input for reducing the tasks. Each output and input of the roles are keep within the filing system. owing to parallel computing nature of Map cut back, parallelizing information mining algorithms exploitation the Map Reduce model has received important attention from the analysis community since the introduction of the model by Google. The Map cut back model supported Hadoop is examined for pertinence within the field of knowledge Mining. Steps to implement Enhanced proposed method with Map Reduce framework:-

Step 1: Maps the input dataset to N partitions, where N = number of slave machines.

Step 2: Reduce phase would take the immediate key-value pairs emitted in the map phase.

Step 3: Send them altogether to the master node for further collecting the number for count per item.

3.1 First, to generate frequent itemset in form of <item, count> key-value pair, which tells the number of occurrences.

3.2 Second, to generate the candidate sets from the source data file. It first prune those items that occur minimum than the support threshold by looking up the global Hash map list and then recursively call GenerateList() function.

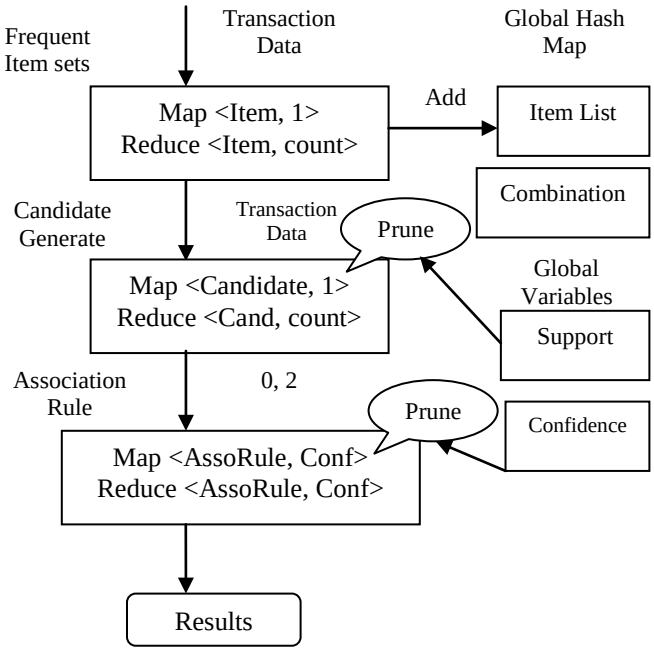


Figure 2: Block diagram to show working of Global Hash in Map Reduce.

7.0 CONCLUSIONS

Association Rule primarily based parallel information mining rule that deals with Hadoop Map Reduce. With this speedy detonation of information, process is preceded from terabytes era to pebibytes era. This trend produces the demand for progression advancement in data collection and storing technology. Thus there's a growing demand to run data processing rule on terribly giant datasets. Hadoop is that the computer code model framework for writing sensible applications that quickly method large amounts of information in parallel on immense clusters of computed nodes. It works on Map Reduce programming model. Map cut back could be a generic execution engine that parallelizes computation over an outsized cluster of machines.

8.0 FUTURE WORK

While experimenting with huge clusters using Hadoop, I came to a problem assertion that it will not disseminate any global variables to be apportioned by every partition due to its nature Sharing-nothing architecture. It can be implemented with Map reduce framework to provide more flexible development environment.

9.0 ACKNOWLEDGMENT

Authors express their deep sense of gratitude to the Founder president of Amity Universe, Dr. Ashok K. Chauhan for his keen interest in promoting research in the Amity University and has always been an inspiration for achieving great heights.

REFERENCES

[1]. Souptik Datta, Kanishka Bhaduri, Chris Giannella, Ran Wolff, and Hillol Kargupta, Distributed Data Mining in Peer-to-Peer Networks, University of Maryland, Baltimore County, Baltimore, MD, USA, Journal Ieeeinternet Computing chronicle Volume 10 Issue 4, Pages 18 - 26, July 2006.

[2]. ning Chen, Nuno C. Marques, and Narasimha Bolloju, A Web Service based methodology for information mining in dispersed situations, Department of Information Systems, City University of Hong Kong, 2005.

[3]. mafruz Zaman Ashrafi, David Taniar, and Kate A. Smith, A Data Mining Architecture for Distributed Environments, pages 27-34, Springer-Verlaglondon, UK, 2007.

[4]. grigoriostsoumakas and Ioannisvlahavas, Distributed Data Mining oflarge Classifier Ensembles, SETN-2008, Thessaloniki, Greece, Proceedings,companion Volume, pp. 249-256, 11-12 April 2008.

[5]. vudasreenivasarao, Multi Agent-Based Distributed Data Mining: Anover View, International Journal of Reviews in figuring, pages 83-92,2009.

[6]. p.kamakshi, A.vinayababu, Preserving Privacy and Sharing the Datain Distributed Environment utilizing Cryptographic Technique on Perturbeddata, Journal

Of Computing, Volume 2, Issue 4, ISSN 21519617, April2010.

[7]. feng LI, Jin MA, Jian-hua LI, Distributed unnamed information perturbationmethod for protection safeguarding information mining, Journal of Zhejiang Universityscience An ISSN 1862-1775, pages 952-963, 2008.

[8]. goswami D.n. et. al., An Algorithm for Frequent Pattern Mining Based On Apriori (IJCSE) International Journal on Computer Science andengineering Vol. 02, No. 04, 942-947, 2010.

[9]. marcingorawski and Paweljureczek, Using Apriori-like Algorithms for Spatio-Temporal Pattern Queries, Silesian University of Technology,institute of Computer Science, Akademicka 16, Poland, 2010.

[10]. cheng-Tao Chu et. al., Map-Reduce for Machine Learning on Multicore,cs Department, Stanford University, Stanford, CA, 2006.

[11]. navrajchohanet. al., See Spot Run: Using Spot Instances for Map-Reduce Workflows, Computer Science Department, University of California,2005.

[12]. Data Mining Textbook: Jawai Hen, and Michael Kamber.