ISSN (ONLINE): 2349-7084
GLOBAL IMPACT FACTOR 0.238
ISRA JIF 0.351

INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING IN RESEARCH TRENDS
VOLUME 2, ISSUE 5, MAY 2015, PP 359-364

Pattern Finding in Large Datasets with Big Data Analytics Mechanism

[1]Y.Usha Sree,[2]P.Ragha Vardhani

Abstract: One of the most popular knowledge discovery approaches is to find frequent items from a transaction data set and derive association rules. Pattern finding is one of the most computationally expensive steps in large data sets. Patterns often referred to association rules. Association rule plays an important role in the process of mining data for sequential pattern. Association rules are used to acquire interesting rules from large collections of data which expresses an association between items or sets of items. . Apriori is a classic algorithm for learning association rules. It is designed to operate on databases containing transactions. Apriori algorithm attempts to find subsets which are common to at least a minimum number of the item sets. Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time and groups of candidates are tested against the data. The algorithm terminates when no further Successful extensions are found. In this paper we enhance Apriori algorithm to solve its complexity over large data sets. Thus in order to address the pattern finding in large datasets we used Big data analytics mechanism it is a collection of large amount of data from numerous sources and usable to be processed at much higher frequency. We first collect variety of data and then integrate both structured and unstructured data using MapReduce to find out sequential pattern from the required data sets.

Keywords: Big Data, MapReduce; Apriori; Association Rule; Pattern mining; Variety of data, Knowledge discovery, Data mining

I. INTRODUCTION

Knowledge discovery is the computer assisted processes which are used to analyze large sets of data and to extract the meaning of those data. Knowledge discovery uses sophisticated mathematical algorithms to segment the data and evaluate the probability of future events. It is the practice to discover hidden patterns and unexpected trends in the data using a combination of techniques from machine learning statistics and database technologies automatically from very large data storage. It involves in cleaning and integrating data from data sources like databases, flat files, pre-treatment of selecting and transferring target data, mining the required knowledge and finally evaluate and present the knowledge [1]

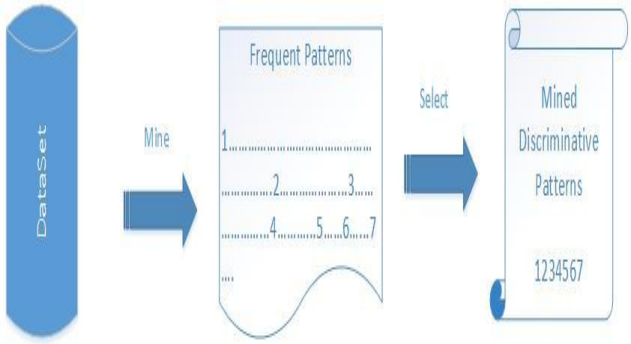

Fig 1. Mine frequent patterns

- ***Y.Usha Sree*** *is currently pursuing master's degree program in computer science & engineering in Ravindra College of Engineering for Women(JNTUA), India,**E-mail: ushasree.16@gmail.com***
- ***P.Ragha Vardhani****,.Currently she is working as Assistant Professor in Department of Computer Science Engineering in Ravindra College of Engineering for Women, Kurnool. India,* ***E-mail: ragha.vardhani@gmail.com***

1.2 ASSOCIATION RULE

Association is a knowledge discovery function that discovers the co-occurrence of items in a collection. The relationships between co-occurring items are expressed as association rules. Association rules are created by analyzing data for frequent if or then patterns and using the criteria support and confidence to identify the most important relationships. Support is an indication of how frequently the items appear in the database. Confidence indicates the number of items if or then statements have been found to be true. An association rule has two parts, an antecedent (if) and a consequent (then). An antecedent is an item found in the data. A consequent is an item that is found in combination with the antecedent. In knowledge discovery, association rules are useful for analyzing and predicting customer behavior. They play an important part in shopping basket data analysis, product clustering, and catalog design and store layout.

1.3 FREQUENT PATTERN MINING

Frequent pattern mining is an essential step in the process of association rule mining and has been a focused theme in knowledge discovery research for over a decade. Frequent patterns are item sets, subsequences, or substructures that appear in a data set with frequency no less than a user-specified threshold. For example, a subsequence, such as buying first a PC, then a digital camera and then a memory card, if it occurs frequently in a shopping history database, is a (frequent) sequential pattern. A substructure can refer to different structural forms, such as subgraphs, subtree or sublattices which may be combined with itemsets or subsequences. If a substructure occurs frequently in a graph database, it is called a (frequent) structural pattern. Finding frequent patterns plays an essential role in mining associations, correlations and many other interesting relationships among data. Moreover, it helps in data indexing, classification, clustering and other knowledge discovery task

as well. Thus, frequent pattern mining has become an important knowledge discovery task and a focused theme in database community

1.4 APRIORI

In knowledge discovery, Apriori is a classic algorithm for learning association rules. Apriori is designed to operate on databases containing transactions. It is proceeds by indentifying the frequent individual items in the database and extending them to larger item sets as long as those item sets appear sufficiently often in the database. In the given sets of itemsets, the algorithm attempts to find subsets which are common to at least a minimum number of the itemsets. Apriori uses a bottom up approach, where frequent subsets are extended one at a time this step known as Candidate generation in which group of candidates are tested against the data. Apriori algorithm terminates when no further successful extensions are found.In this paper we describe the improved Apriori algorithm based on MapReduce mode, which can handle massive datasets with a large number of nodes on Hadoop platform.

1.4.1 PSEUDOCODE FOR APRIORI ALGORITHM

Join Step: Ck is generated by joining Lk-1 with itself

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

1. Ck: Candidate itemset of size k

2. Lk: frequent itemset of size k

3. L1= {frequent items};

4. for(k=1; Lk!=0; k++) do begin

5. Ck+1= candidates generated from Lk;

6. for each transaction T in database do increment count of all candidates in Ck+1 that are contained in T

7. Lk+1= candidates in Ck+1 with min_support 8. end 9. return UkLk; 2

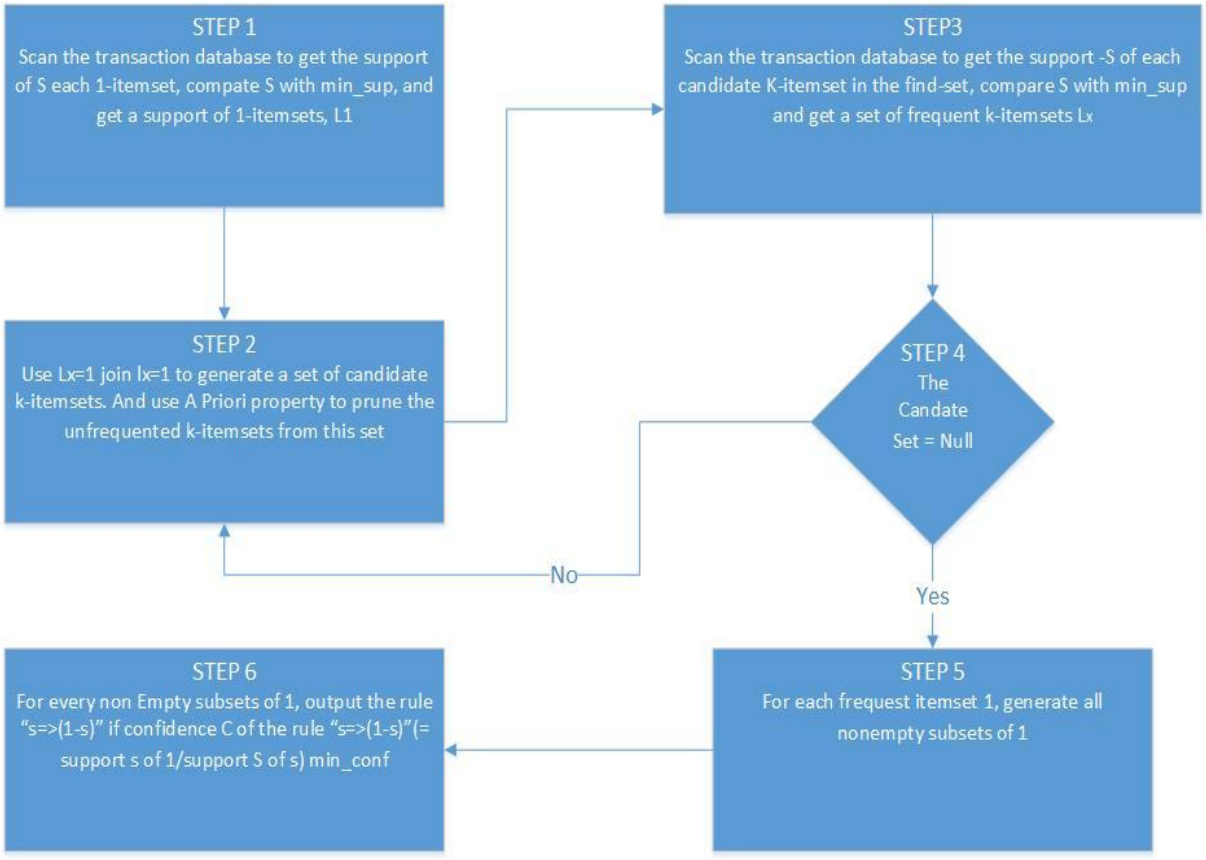

Fig 2. Steps to Process APRIORI

ISSN (ONLINE): 2349-7084

INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING IN RESEARCH TRENDS
VOLUME 2, ISSUE 5, MAY 2015, PP 359-364

II. TYPES OF BIG DATA AND SOURCES:

There are two types of big data: structured and unstructured. Structured data are numbers and words that can be easily categorized and analyzed. These data are generated by things like network sensors embedded in electronic devices, smartphones, and global positioning system (GPS) devices. Structured data also include things like sales figures, account balances, and transaction data. Unstructured data include more complex information, such as customer reviews from commercial websites, photos and other multimedia, and comments on social networking sites. These data can not easily be separated into categories or analyzed numerically. “Unstructured big data is the things that humans are saying,” says big data consulting firm vice president Tony Jewitt of Plano,Texas. “It uses natural language.”Analysis of unstructured data relies on keywords, which allow users to filter the data based on searchable terms. The explosive growth of the Internet in recent years means that the variety and amount of big data continue to grow. Much of that growth comes from unstructured data.

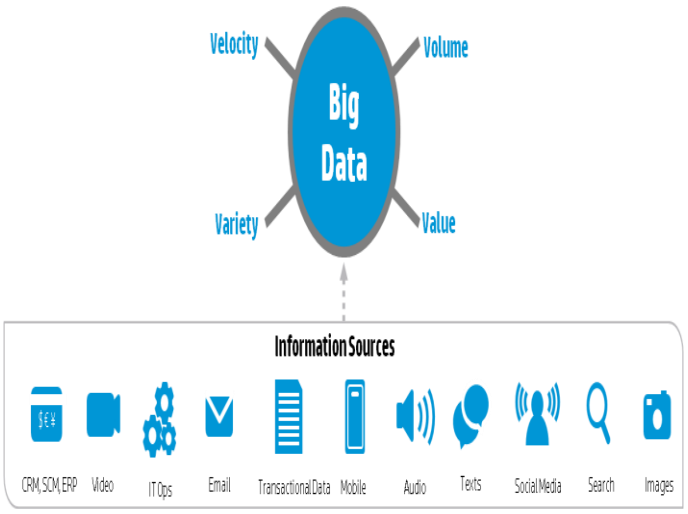

Fig 3. Big data Sources

III. BIG DATA ANALYTICS TOOLS

There are varieties of applications and tools developed by various organizations to process and analyze Big Data. The Big Data analysis applications support parallelism with the help of computing clusters. These computing clusters are collection of hardware connected by Ethernet cables. The following are major applications in the area of Big Data analytics.

i. MapReduce

The framework is divided as following:

Map: A function that parcels out work to different nodes in the distributed cluster

Reduce: A function that collects the work and resolves the results into a single value.

MapReduce framework is fault-tolerant because each node in the cluster is expected to report back periodically with completed work and status updates. If a node remains silent for longer than the expected interval, a master node makes note and re-assigns the work to other nodes.

As a data parallel model, MapReduce is a patented software framework introduced by Google to support distributed computing on large datasets on clusters of computers [1]. Known as a simple parallel processing mode, Map-reduce has many advantages: such as, it is easy to do parallel computation, to distribute data to the processors and to load balance between them, and provides an interface that is independent of the backend technology [10]. MapReduce is designed to describe the process of parallel as Map and Reduce. The user of the MapReduce library expresses the computation as two functions: Map and Reduce [2]. Map, written by the user, takes an input pair and produces a set of intermediate key/value pairs. The MapReduce library groups together all intermediate values associated with the same intermediate key I and passes them to the Reduce function. The Reduce function, also written by the user, accepts an intermediate key I and a set of values for that key. It merges together these values to form a possibly smaller set of values. Typically just zero or one output value is produced per Reduce invocation. The intermediate values are supplied to the user's reduce function via iterator. This allows us to handle lists of values that are too large to fit in memory. The Map and Reduce functions are both defined with respect to data structured in (key, value) pairs. MapReduce provides an abstraction that involves the programmer defining a “mapper” and a “reducer”, with the following signatures [9]: Map::(key1) ⇒ list(key2,value2) Reduce::(key2,list(value2) ⇒ list(value2)

MapReduce is a programming model for computations on massive amounts of data and an execution framework for large-scale data processing on clusters of commodity servers. It was originally developed by Google and built on well-known principles in parallel and distributed processing [15]. MapReduce program consists of two functions –Map function and Reduce function. MapReduce computation executes as follows

1. Each Map function is converted to key-value pairs based on input data. The input to map function is tuple or document. The way key-value pairs are produced from the input data is determined by the code written by the user for the Map function

2. The key-value pairs from each Map task are collected by a master controller and sorted by key. The keys are divided among all the Reduce tasks, so all key-value pairs with the same key wind up at the same Reduce task.

3. The Reduce tasks work on one key at a time, and combine all the values associated with that key in some way. The manner of combination of values is determined by the code written by the user for the Reduce function.

MapReduce has two major advantages:

The Map Reduce model hide details related to the data storage, distribution, replication, load balancing and so on. Furthermore, it is so simple that programmers only specify two functions, which are map function and reduce function, for performing the processing of

ISSN (ONLINE): 2349-7084

INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING IN RESEARCH TRENDS
VOLUME 2, ISSUE 5, MAY 2015, PP 359-364

the Big Data. Map Reduce has received a lot of attentions in many fields, including data mining, information retrieval, image retrieval, machine learning, and pattern recognition.

Fig 4. MapReduce logical dataflow

IV. PROBLEM JUSTIFICATION

Frequent Pattern Mining is an important knowledge discovery task and it has been a focus theme in knowledge discovery research. Frequent Pattern Mining over large database is fundamental to many knowledge discovery applications. One of the main issues in Frequent Pattern Mining is Sequential Pattern Mining retrieved the relationships among objects in sequential dataset[2]. Apri ori All is a typical algorithm to solve the problem in Sequential Pattern Mining but its complexity is so high and it is difficult to apply in large datasets. Recently, to overcome the technical difficulty, there are a lot of researches on new approaches as follow

1. Custom built Apriori algorithm
2. Modified Apriori algorithm
3. Frequent Pattern-tree and its development
4. Integrating Genetic algorithms
5. Rough set Theory/ Dynamic Functions

4.1 PROBLEMS OF FREQUENT PATTERN MINING

In frequent Pattern Mining there are huge number of frequent itemsets which are hard to analyze and most of them are similar.

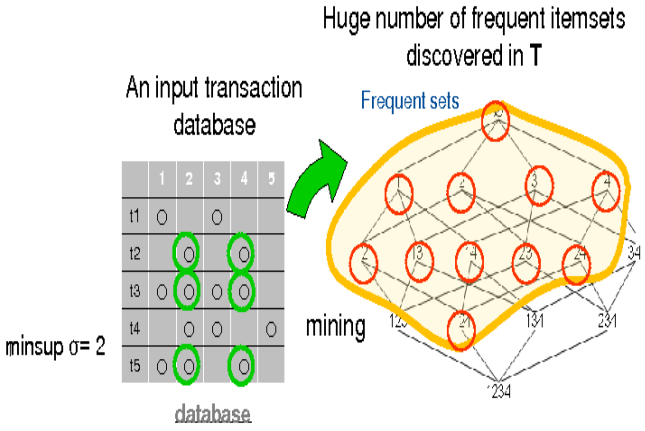

Fig 5. Problems of Frequent Pattern Mining

V. APPROACHES FOR PATTERN MINING

Following are the approaches of pattern mining

5.1 APRIORI-BASED METHOD The Apriori property of sequences states that, if a sequences S is not frequent, then none of the super-sequences of S can be frequent[3] . E.g. is infrequent implies that its super-sequences like and would be infrequent too. The GSP algorithm finds all the length-1 candidates(using one database scan) and orders them with respect to their support ignoring ones for which support < min_sup. Then for each level (i.e., sequences of length-k), the algorithm scans database to collect support count for each candidates sequence and generates candidate length-(k+1) sequences from length-k frequent sequences using Apriori. This is repeated until no frequent sequence or no candidate can be found. Advantages Uses large itemset property.

- Easy to implement.
- Easily parallelized.ϖ Disadvantages Large number of candidates sets are generated.
- Multiple database scans are required.
- The system I/O cost increases due to multiple scanning of transactional database.

5.2 VERTICAL FORMAT-BASED METHOD

This is a vertical format sequential pattern mining method. SPADE first maps the sequence database to a vertical id-list database format which is a large set of items. Sequential pattern mining is performed by growing the subsequences (patterns) one item at a time by Apriori candidate generation. Advantages Reduces the time of scanning candidate sets.

Hash structure is used for storage of candidate sets.ϖ Disadvantages As the minimum confidence value is increased, the time is also increased.

5.3 PATTERN GROWTH BASED METHOD Pattern growth based method uses frequent items to recursively project sequence database into a set of smaller projected databases and grows subsequence fragments in each projected database. This process partitions both the data and the set of frequent patterns to be tested and confines each test being conducted to the corresponding smaller projected database. It first scans the database, collects the support for each item and finds the set of frequent items. Advantages Decreases the system overhead.

- Reduces the running time.
- Disadvantage Multiple database scans required.

5.4 CONSTRAINT BASED METHODS

In sequential pattern mining algorithm users can specify only min_sup as a parameter. There are two major difficulties in sequential pattern mining: Effectiveness: The mining may return a huge number of patterns, many of which could be uninteresting to users. Efficiency: It often takes substantial computational time and space for mining the complete set of sequential patterns in a large sequence database. To prevent these problems, users can use constraint based sequential pattern mining for focused mining of desired patterns. Constraints could be anti-monotone, monotone, succinct, convertible or inconvertible. Anti-monotonicity means "if an item-set does not satisfy the rule constraint, then none of its supersets satisfy". Monotonicity means "if an item-set satisfies the rule constraint, then all of its supersets satisfy". Succinctness means "All and only those patterns guaranteed to satisfy the rule can be enumerated". Convertible constraints are those which are not any of anti-monotonic, monotonic, succinct but can be made anti-monotonic or monotonic constraints by changing order of elements in the set. Inconvertible constraints are the ones which are not convertible.

Advantage

- Decreases the system overhead.
- Easily parallelized

Disadvantage

The system I/O cost increases due to multiple scanning of transactional database

VI. FUTURE WORK

Ideas for future work in pattern mining include: Applying telescoping in tree projection pattern-growth algorithm to reduce the tree size, where more than one item can be compressed into one node or edge. Distributed mining of sequences can provide a way to handle scalability in very large sequence databases and long sequences. In the area of web usage mining, this can be applied to mine several web logs distributed on multiple servers. Extending the capability of existing approaches.

VII. CONCLUSION

In this paper, we explore pattern mining and its algorithm which are used to find patterns by integrating structured and unstructured data together using MapReduce framework. Sequential pattern mining methods have been used to analyze this data and identify patterns. Such patterns have been used to implement efficient systems that can recommend based on earlier pragmatic patterns, help in making predictions, to improve usability of system, detect events and in general help in making strategic product decisions. We have also seen applications of pattern mining in variety of domain. Apart from this, new sequential pattern mining methods may also be developed to handle special scenarios of colossal patterns, approximate sequential patterns and other kinds of sequential patterns specific to the applications

REFERENCES

[1] Assured Research, Essay on Big Data. June 2012.

[2] Bart Goethals. Survey on Frequent Pattern Mining.

[3] Basel Kayyali, David Knotl, Peter Crroves and Steve Van Kuiken. The Big Data

[4] Bid Data, ANew World of Opportunities. NESSI White Paper, December 2012.

[5] Albert Bifet, Mining Big Data in Real Time. Informatica 37, 15-20 (2013).

[6] Amit Ganatra, Amit Thakkar and Cletna Chand. Sequential Pattern mining: Survey and Current Research Challenges. International Journal of Soft Computing and Engineering, Volume 2, March 2012.

[7] Aniket Mahanti and Reda Alhajj. Visual Interface for Online Watching of Frequent Itemset Generation in Apriori and Eclat, Fourth International Conference on Machine Learning and Application (2005).

[8] Anuradha Sharma, Arvind Sehwal and Harleen Puri. An Empirical Proposal towards the Algorithmic Approach and Pattern in Web Mining for Assorted Applications. International Journal of Innovative Research in Computer and Communication Engineering, Volume 1. April 2013.

[9] Aramburu, Juan Manuel Perez, Maria Jose, Rafael Berlanga and Torben Bach Pedersen, Integrating Data warehouse with Web Data: A Survey, IEEE transaction on knowledge and Data Engineering, Volume 20, July 2008.

[10]Big Data Strategy- Issues Paper. March 2013. [11]C.I. Ezeife, R. Mabroukeh and Nizar. A Taxonomy of Sequential Pattern Mining Algorithms. ACM Computing Surveys, Volume 43, November 2010. [12]Challenges and Opportunities with Big Data, A Community white paper developed by leading researchers across the United States.

[13]D. Magdalene Delighta Angeline and I. Samuel Peter James. Association Rule Generation using Apriori Mend Algorithm for Student's Placement. Int. J. Emerging. Science. 2(1). March 2012.

[14]D.P Agrawal, R.K. Gupta and A. Tiwari. A Survey on Frequent Pattern Mining: Current Status and Challenging Issues. Information Technology Journal 9(7), 2010.

[15]David Loshin, Integrating Structured and Unstructured Data. TDWI Checklist Report.

[16]DonXin, Hong Cheng, Jiawei Han and Xifeng Yan. Frequent Pattern Mining : Current Status and Future direction. Knowledge discovery Knowledge Disc, 2007.

[17]Dr.L. DE Radt, Mining patterns in Structured Data. September 2009.

[18]Ekta Garg and Meenakshi Bansal. A Survey on Improved Apriori Algorithm. IJERT, Volume 2, July 2013.

[19]Huajun Chen, Jun Ma , Xiangyu Zhang, Xiaolongyu and Yang Liu. Map Reduce- Based Pattern Finding Algorithm Applied in Motif Detection for Prescription Compatibility Network, APPT 2009.

[20]Hui Xion, Jian Pei and Yan Huang. Mining Co-Location Patterns with Rare Events from Spatial Data Sets. Geoinformatica (2006).

INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING IN RESEARCH TRENDS
VOLUME 2, ISSUE 5, MAY 2015, PP 359-364

ABOUT AUTHORS

Y.USHA SREE received B,Tech degree in Computer Science and Engineering from Madanapalli Institute of Technology and Science (JNTUA)and now pursuing M.Tech (CSE) from Ravindra College of Engineering for Women(JNTUA). Research interest includes Big Data and Data Mining.

Mrs. P.RAGHA VARDHANI, received B.Tech (CSE) degree from Vaagdevi Institute of Technology & Science and M.Tech (CSE) from Vaagdevi Institute of Technology & Science. Currently she is working as Assistant Professor in Department of Computer Science and Engineering in Ravindra college of Engineering for Women, Kurnool. She has lifetime membership in Indian Institute of Technical Education (ISTE).