IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-5, Issue-6 E-ISSN: 2347-2693

Infrequent Weighted Itemset Mining for Large Dataset

R.B.M. Sayyad[1*] and P.S. Yalagi[1]

[1*]Department of CSE, Walchand Institute of Technology (Solapur University), Solapur, India
[2]Department of CSE, Walchand Institute of Technology (Solapur University), Solapur, India

[*]*Corresponding Author: sayyadriz@gmail.com*

Available online at: www.ijcseonline.org

Received: 12/May/2017, Revised: 24/May/2017, Accepted: 16/Jun/2017, Published: 30/Jun/2017

***Abstract*—** Data mining is the process of analysing data from many different perspectives or dimensions, categorize it and finally summarize it into useful information. This information can be used to increase profits, cut costs, or both. Data mining software is used for analysing data. It allows users to analyse data from many different perspectives, categorize it, and summarize the relationships discovered. Specially, data mining is the way of extracting valuable correlations or patterns among many number of fields in large relational databases. Pattern mining has become an important task in data mining. Mining frequent and infrequent itemsets from a dataset is the most important field of data mining. Mining frequent itemset is very expensive when minimum support threshold is low, and when a minimum support threshold is high mining in frequent itemsets is highly expensive. The proposed system uses multiple level minimum supports to constrain infrequent itemsets by giving different minimum supports to itemsets with different length in order to mine a number of infrequent itemsets in an appropriate degree. In this paper, we are implementing the concept of infrequent weighted itemset mining based on Hadoop-MapReduce model, which can handle massive datasets in mining in frequent itemsets, in that we proposed two novel algorithms based on IWI Miner, IWI Miner to drive the IWI mining process. This paper emphasis on the issue of discovering those itemsets which occurs rarely in large dataset called infrequent weighted itemset (IWI) mining problem.

***Keywords*—** *Data Mining, frequent Itemset, Infrequent Itemset, Weighted Itemset, Hadoop , MapReduce*

I. INTRODUCTION

Data Mining is the process of discovering interesting and important knowledge from large amounts of data. This knowledge can be used to increase profits or to reduce costs or both. Itemset mining is one of the important data mining technique used for discovering the valuable correlation or patterns among data in the dataset. The first attempt was focused on discovering frequent itemsets. An itemset is frequent if its support count is greater than or equal to a user-specified threshold, called minimum support(ms) otherwise the itemset is infrequent. Infrequent itemsets are often considered to be uninteresting and are eliminated using the support measure. Although the majority of infrequent itemsets are uninteresting, some of them might be useful to the analysis. Some infrequent itemsets may also shows the occurrence of rare events which can be interesting or exceptional situations in the data.

However, many traditional approaches only consider whether an item is present in a transaction, but does not take into account the weight of an item within a transaction. In order to consider items/transactions differently, the notion of weighted itemset has been introduced. A weight is associated with each data item within a transaction to reflect the interest of the item within the transaction.

Recently there has been increasing demand for mining infrequent weighted itemset mining. Traditional infrequent itemset mining algorithms suffer from scalability when dataset is large. So, single processor's memory and CPU resources are not sufficient to handle huge datasets. Solution to the above problem is distributed computing. Distributed data mining algorithms attempt to divide the mining task into subtasks and assign each task to one homogenous node such that each node works simultaneously and independently. There are few issues raises in distributed data mining like data partitioning, load balancing, communication cost, identification of failure of nodes. To solve the above issues, the Map, reduce framework has been introduced. Map-reduce is a distributed framework more suitable for data processing. Hadoop MapReduce is a framework for easily writing applications for distributed processing of large volume of data on large clusters of commodity hardware in a reliable, fault-tolerant manner. Here cluster can be of thousands of nodes and dataset can be in multi TBs. Users shows the computation in terms of a map and a reduce function clearly, and the underlying execution system automatically parallelizes the computation across a number of homogenous machines. This model also handles machine failures, and schedules, communication between machines to make efficient use of the network and disks

II. LITERATURE REVIEW

Itemset mining is one of the important technique in data mining widely used for discovering the valuable correlation among data. Most of the research work has been done on frequent itemset mining because the majority of infrequent itemsets are uninteresting, some of them might be interesting and that can be useful to the analysis.

In the previous research, different algorithms are used for mining frequent and infrequent itemsets from datasets based on different platforms.

Comparison between different algorithms for mining Infrequent itemsets from large datasets. IWIM i.e. Infrequent Weighted Itemset Mining and MIWIM i.e. Minimal Infrequent weighted Itemset Miner are the two algorithms most widely used for mining infrequent itemsets. The comparison between various itemset mining algorithms is in terms of the type of itemsets, performance, Drawbacks. Comparative analysis indicates that IFP min i.e. infrequent pattern mining is faster than MINIT i.e. Minimal Infrequent Itemset when min. Support is high. IFP min and MIWI i.e. Minimal Infrequent Weighted Itemset mining algorithm work superior than MINIT for every support. When min. Support threshold is low MIWI works better than IFP and MINIT. The running time of MIWI grows when IWI support min is medium. The running time of IWI miner rises when IWI support min is high. When IFP_MLMS is compared with Apriori_MLMS then execution time of both algorithm increases when the number of transactions increases. The growth rate of Apriori_MLMS algorithm is larger than IFP_MLMS in terms of execution time. Apriori MLMS fails to give results for large datasets so IFP_MLMS performs better than Apriori_MLMS. [1].

Several algorithms have been proposed for discovering rare and weighted itemsets, i.e., the Infrequent Weighted Itemset (IWI) mining problem. Two novel algorithms, i.e. IWI and Minimal IWI miner have been developed to drive the IWI mining process. This algorithm considers the weight associated with each data item within a transaction to reflect the interest of the item within the transaction [2]. When the dataset size is very large, single processor's memory and CPU resources are not sufficient to handle very large datasets. To handle very large datasets Parallel and distributed computing is effective to use. A method based on Hadoop-MapReduce model, which can handle massive datasets in mining infrequent item sets. Hadoop MapReduce is a framework for easily writing applications for distributed processing of large volume of data on large clusters of commodity hardware in a reliable, fault-tolerant manner. Here cluster can be of thousands of nodes and dataset can be in multi TBs. Users shows the computation in terms of a map and a reduce function clearly, and the underlying execution system automatically parallelizes the computation across a number of homogenous machines. This model also handles machine failures, and schedules, communication between machines to make efficient use of the network and disks [3].

Mining frequent itemset is very expensive when minimum support threshold is low, and when a minimum support threshold is high mining infrequent itemsets is highly expensive. So, multiple level minimum supports are used to discover infrequent itemsets by giving different minimum supports to itemsets with different length in order to mine number of infrequent itemsets in an appropriate degree. Apriori_MLMS algorithm to mine both frequent and infrequent itemsets simultaneously. This work did not focus on generating positive and negative association rules from the discovered frequent and infrequent itemsets [4].

Weighted Negative Association Rules from Infrequent Itemsets Based on Multiple Supports(WNAIIMS) is used to discover Weighted Negative Association Rules from Infrequent Itemsets Based on Multiple Supports. This technique allows users to specify multiple minimum supports to reflect their varied frequencies in the database. Existing algorithms ignores negative association rules from infrequent itemsets. Furthermore, this technique sets different weighted values for items according to the importance of each item [5].

Pattern-Growth Paradigm and Residual Trees is the novel concept that can be used to discover Minimal Infrequent Itemsets. An itemset is a minimally infrequent itemset (MII) if and only if all its subsets are frequent. Thus, a trivial algorithm to mine all MIIs would be computationally expensive. The use of residual trees reduces the time required for execution. A residual tree for a specific item is a tree representation of the residual database corresponding to the item. The residual database corresponding to an item is the database of transactions obtained by removing that item from Transaction database. The IFP-tree corresponding to this database is called residual tree [6]. MapReduce Framework is used for Mining Interesting Infrequent Itemsets from Very Large Data. An Algorithm based on MapReduce uses two phases in order to mine infrequent itemsets. The results show that the proposed method is very efficient in finding infrequent items from very large datasets as the size of dataset increases. If mappers generate more intermediate results during the mining process, then performance of reducers will reduce [7].

III. SYSTEM DESIGN

A. TECHNOLOGY USED

1. Traditional Approach

In the traditional approach as shown in Fig1.Traditional Approach, an organization will have a personal computer to store and process big amount of data. Here data will be stored in an RDBMS like Oracle Database, MS SQL Server and sophisticated software can be written to connect with the database. This database processes the required data and present it to the users for analysis purpose. This approach

performs better when we have less volume of data that can be adapted by standard database servers, or depends on the

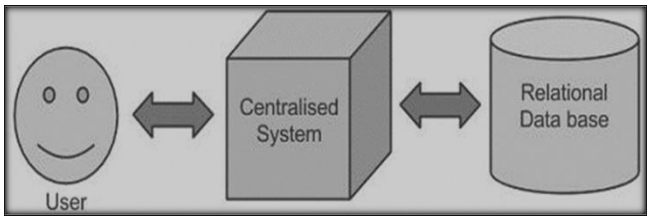

Fig1.Traditional Approach

processing limit of the processor which is processing the data. But when we have a huge volume of data, it is really a tedious task to process such data through a traditional database server.

2. Hadoop

When we have a huge volume of data, it is really a tedious task to process such data in order to mine infrequent weighted itemset through a traditional database server because single processor's memory and CPU resources are not sufficient to handle very large datasets for mining infrequent weighted itemsets that's why the proposed system uses a method based on Hadoop MapReduce model, which can handle massive datasets in mining infrequent weighted itemsets. Hadoop is an Apache open source framework written in Java that supports distributed processing of large datasets across clusters of nodes using simple programming models. A Hadoop is a frame worked application works in an environment that offers distributed storage and computation across large clusters of computers. Hadoop is designed to maximize from a single server to thousands of machines, each machine offers its local computation and storage. Hadoop MapReduce divides the task into small chunks and assigns those chunks to many computers connected to the network, and gathers the results to form the final result dataset. The proposed system takes input as a transactional dataset this input is partitioned into a number of splits and assign each split and Multilevel threshold to MapReduce Phase1 to generate a list of weighted supports of items. Result of phase 1 will be stored on the Hadoop Distributed File System HDFS). The result of phase 1 and multilevel threshold will be given as input to MapReduce phase2.This phase generates FP-tree that will be given as input to IWI mining algorithms that will find Infrequent weighted itemsets.

a. Hadoop Architecture

The Hadoop framework consists following four modules:

- Hadoop Common: Some of the Hadoop modules need Java libraries and utilities. Hadoop Common provides these Java libraries and utilities. These libraries offer file system and OS level abstractions and holds some necessary Java files and scripts required to start Hadoop.
- Hadoop YARN: This module is for handling job scheduling and cluster resource management.
- Hadoop Distributed File System (HDFS™): A distributed file system that offers high throughput access to application data.
- Hadoop MapReduce: This module is used for parallel processing of large data sets.

b. MapReduce

Hadoop MapReduce is a software framework for easily writing applications which process a large volume of data in parallel on large clusters in a reliable, fault-tolerant manner. Each cluster consists of thousands of nodes & each node has its own local resources.

The term MapReduce refers to the following two distinct tasks that Hadoop programs perform (As shown in Fig2. MapReduce structure):

- Map stage: On map stage the map or mapper's task is to process the input data. Generally, the input data is in the form of a file or directory and is stored in the Hadoop Distributed File System (HDFS). The input file, i.e. file or directory is passed line by line to the mapper function. The mapper processes the input data and creates its several small parts.
- Reduce stage: This stage is the combination of the Shuffle stage and the Reduce stage. The Reduce or Reducer's task is to process the data that is received from the mapper. After processing, it generates a new set of outputs, which will be stored in the Hadoop Distributed File System (HDFS).

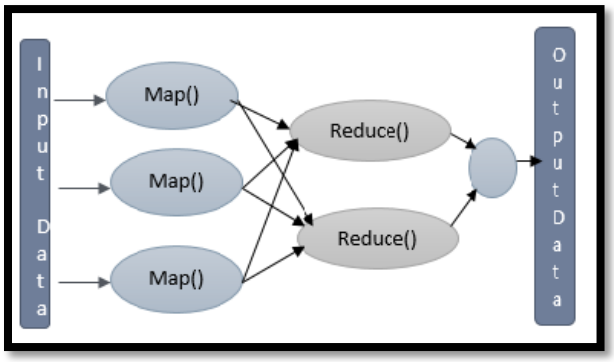

Fig2. MapReduce Structure

c. HDFS Architecture

HDFS holds a very large amount of data and provides easier access to a user. To store such huge amount of data, the files are stored across multiple machines. HDFS is reliable, highly fault tolerant and designed using low-cost hardware.

HDFS follows the master-slave relationship. The system having the name node acts as the master server. Each node in a cluster has a data node. These data nodes manage the data storage of their system. Generally, the user data is stored in the files of HDFS. The file in a file system will be divided into one or more chunks, and stored in individual data nodes. The collection of chunks is called as blocks.

B. SYSTEM ARCHITECTURE

When the dataset size is very large, single processor's memory and CPU resources are not sufficient to handle very large datasets. Parallel and distributed computing is essential approaches to managing very large datasets. The proposed system uses a method based on Hadoop MapReduce model, which can handle massive datasets in mining infrequent item sets. Hadoop MapReduce is a software framework for easily writing applications which process vast amounts of data in parallel on large clusters of commodity hardware in a reliable, fault-tolerant manner. The dataset may be in multi TBs in size. A MapReduce job partitions the input data which may be in the form of file or directory into independent chunks which are processed by the map tasks in a completely parallel and independent manner.

The framework sorts the outputs of the maps, which are then given as input to the reduce tasks. Generally, both the input and the output of the job are stored in a file system. The framework takes care of scheduling tasks, monitoring them and re-executes the failed tasks.

\

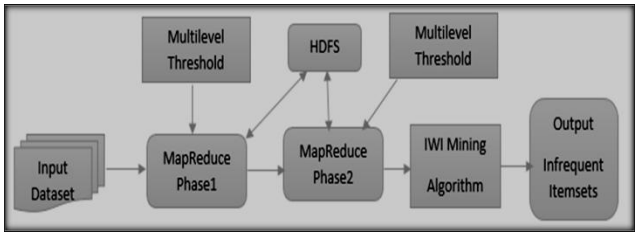

Fig3.Simple Flow of Proposed Work

As shown in the Fig3.Simple Flow of Proposed Work. In that input is a transactional dataset this input is partitioned into a number of splits and assign each split and Multilevel threshold to MapReduce Phase1 to generate a list of weighted supports of items. Result of phase 1 will be stored on the Hadoop Distributed File System HDFS). The result of phase 1 and multilevel threshold will be given as input to MapReduce phase2.This phase generates FP-tree that will be given as input to IWI mining algorithms that will find Infrequent weighted itemsets.

As shown in Fig4.Flow of proposed work. In that input is a transactional dataset this input is partitioned into a number of splits and assign each split to a mapper node, then, Map1 will find a local weight of each item in that split, then, the output of Map1 will be given as input to Reduce1.Reduce1 will take another two inputs called Max.supp and Min.interest and find local weighted support of an items. And generate infrequent items. Item is infrequent if its support is less than, equal to Max. Supp and Interest is greater than equal to Min. interest. Then the result of Reduce1 will be stored in the Hadoop Distributed File System. Now, Hadoop Distributed File System contains a List of weighted supports of items and input split. List of weighted supports of items and input split in HDFS will be given as input to Map2.Map2 takes another two input, i.e. Max. Supp, Min. interest to find the count of each Itemset in a split. Map2 will find local infrequent weighted itemset. Itemset by comparing itemset's support and interest with Max.supp and Min.interest. Then the result of Map2 will be given as input to Reduce2.Reduce2 will create a FP tree for pruning nodes, which are never belong to itemsets that satisfy Maximum support threshold. Then result of Reduce2 will be given as input to the IWI mining algorithm. IWI mining algorithm produces infrequent weighted itemsets. As shown below Flow of Proposed work.

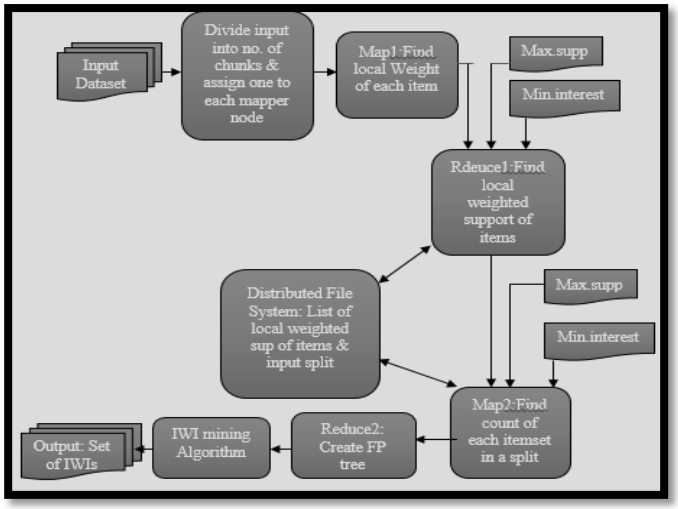

Fig4.Flow of Proposed work

IV. IMPLEMENTATION

The below steps give an overview about infrequent weighted itemset mining.

Algorithm

Input: Dataset, Multilevel Max.supp
Output: Set of IWIS
Step1: Split dataset into no. of chunks & assign one for each mapper node
Step2: Mappers finds local weight of each item
Step3: Reducers find local weighted support of each item
Step4: Result will be stored in HDFS
Step5: Mappers will find count of each itemset
Step6: Reducers will create FPTree for pruning
Step7: IWI mining algorithm generates set of IWIS.

V. EXPECTED RESULT

- The proposed system takes input as a transactional dataset, Maximum Support thresholds and Minimum interest for Mining Infrequent Weighted itemsets.
- Set of Infrequent itemsets will be generated based on multiple maximum support thresholds and minimum utility means those itemsets whose support count is less than equal to multiple maximum support thresholds and whose utility is greater than user specified minimum utility.
- Based on minimum utility we can compare and shows that there can be some infrequent itemsets which produces more benefits compared to some frequent itemsets these itemsets are called interesting infrequent itemsets that will be discovered.

VI. CONCLUSION

Most of the research work has been done on Frequent Itemset Mining but much less attention has been given to mining Infrequent Itemsets. Recently there has been increasing demand for mining infrequent itemsets because Some infrequent itemsets may also suggest the occurrence of interesting rare events or exceptional situations in the data. This paper focuses on mining infrequent itemsets based on multiple maximum support thresholds and minimum utility means those itemsets whose support count is less than equal to multiple maximum support thresholds and whose utility is greater than user specified minimum utility are called interesting infrequent itemsets using that will be discovered. This paper proposes two novel algorithms based on IWI Miner and Minimal Infrequent Itemset Miner using Hadoop. Mining infrequent patterns from large datasets efficiently and the interesting patterns from the discovered patterns is the challenging tasks in the field of infrequent itemset mining. So, this paper uses Hadoop MapReduce Framework to mine infrequent patterns efficiently from large datasets.

ACKNOWLEDGEMENT

I am thankful to my guide Asst. Prof. P. S. Yalagi for her advice and encouragement for the paper work.

REFERENCES

[1] Aruna J. Chamatkar and P.K. Butey , "*Comparison on Different Data Mining Algorithms*", International Journal of Computer Sciences and Engineering, Vol.2, Issue.10, pp.54-58, 2014.

[2] Akilandeswari. S and A.V.Senthil Kumar, "*A Novel Low Utility Based Infrequent Weighted Itemset Mining Approach Using Frequent Pattern*", International Journal of Computer Sciences and Engineering, Vol.3, Issue.7, pp.181-185, 2015.

[3] Jeffery Dean and Sanjay Ghemawat, "MapReduce: simplified data processing on large clusters", Communications of the ACM, Vol. 51, No.1, 2008, pp. 107-113.

[4] Dong, Z Zheng, Z Niu and Q Jiam," Mining infrequent itemset based on multiple level minimum supports", 2nd Int. Conf. on Innovative Computing, Information Control, 2007.

[5] He Jiang, Xiumei Luan, Xiangjun Dong," Mining Weighted Negative Association Rules from Infrequent Itemsets Based on Multiple Supports", 978-0-7695-4792-3/12 $26.00 © 2012 IEEE 2012 International Conference on Industrial Control and Electronics Engineering.

[6] A. Gupta, A. Mittal, and A. Bhattacharya, "Minimally Infrequent Itemset Mining Using Pattern-Growth Paradigm and Residual Trees", Proc. Int'l Conf. Management of Data (COMAD), pp. 57-68, 2011.

[7] T Ramakrishnudu, R B V Subramanyam," Mining Interesting Infrequent Itemsets from Very Large Data based on MapReduce Framework", I.J. Intelligent Systems and Applications, 2015, 07, 44-49.

[8] David J. Haglin and Anna M. Manning, "On Minimal Infrequent Itemset Mining".

[9] K. Sun and F. Bai, "Mining Weighted Association Rules Without Preassigned Weights," IEEE Trans. Knowledge and Data Eng., vol. 20, no. 4, pp. 489-495, Apr. 2008.

[10] Ling Zhou, Stephen Yau ∗," Efficient association rule mining among both frequent and infrequent items", Computers and Mathematics with Applications 54 (2007) 737–749.

[11] J.Jaya1, S.V.Hemalatha2," A Survey of Frequent and Infrequent Weighted Itemset Mining Approaches".

[12] He Jiang, Xiumei Luan, Xiangjun Dong," Mining Weighted Negative Association Rules from Infrequent Itemsets Based on Multiple Supports", 978-0-7695-4792-3/12 $26.00 © 2012 IEEE 2012 International Conference on Industrial Control and Electronics Engineering.

[13] Junfeng Ding, Stephen S.T. Yau, "TCOM, an innovative data structure for mining association rules among infrequent items", Computers and Mathematics with Applications, Vol. 57, No. 2, January 2009, pp. 290-301.

[14] Guru Prasad M.S., Nagesh H.R., Swathi Prabhu, "*An Efficient Approach to Optimize the Performance of Massive Small Files in Hadoop MapReduce Framework*", International Journal of Computer Sciences and Engineering, Vol.5, Issue.6, pp.112-120, 2017.

[15] Nidhi Sethi and Pradeep Sharma, "*Mining Frequent Pattern from Large Dynamic Database Using Compacting Data Sets*", International Journal of Scientific Research in Computer Science and Engineering, Vol.1, Issue.3, pp.31-34, 2013.

Authors Profile

Rizwana Begum Sayyed Mustafa is pursuing her Master Degree of Computer Science & Engineering from Walchand Institute of Technology, Solapur University, Solapur Maharashtra, India. She has received Bachelor degree in Computer Science and Engineering from Matoshri Pratishthan Group of Institutions, SRTM University, Nanded. Her main research work focuses on Data mining, Pattern Mining.

Ms. Pratibha S. Yalagi is working as Assistant Professor in Computer Science and Engineering / Information Technology at Walchand Institute of Technology, Solapur University, Solapur, Maharashtra, India. She is pursuing Ph.D. in distributed and parallel computing. She has 16 years of experience in teaching. She has presented and published more than 25 papers in various national, international conferences and journals. She worked as a Member, Board of Studies, Information Technology, Solapur University, Solapur.

