Novelty Journals

ISSN 2394-7314

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 3, Issue 1, pp: (130-134), Month: January-April 2016, Available at: www.noveltyjournals.com

AN EFFICIENT BIG DATA PROCESSING FOR FREQUENT ITEMSET MINING BASED ON MAPREDUCE FRAMEWORK

[1]K. Jayabal, [2]Dr. P. Marikkannu

[1, 2] Information Technology, Anna University Regional Campus, Coimbatore, India

Abstract: **In modern days, size of datasets becomes larger and larger. The conventional tools and techniques cannot able to handle those large data. In order to extract value from massive amount of data, frequent itemset mining is an important tool which could give better insights for the business environment. Frequent itemset mining is the most popular technique for data mining, but main memory was not capable to load these large datasets. The main goal of frequent itemset mining is to extract frequent itemset MapReduce is used to overcome this limitation by using parallel processing of Big Data which helps to handle problem of large datasets. The proposed ClustBigFIM works based on MapReduce framework for mining large datasets ClustBigFIM is modified BigFIM algorithm providing extensibility and speed in order to reveal useful information from large datasets. ClustBigFIM is the hybrid approach which is the combination of apriori and eclat algorithm. The proposed algorithm uses k-means clustering algorithm for clustering, apriori and eclat to mine frequent itemsets. ClustBigFIM algorithm is used to make effective business decisions in competitive business environment by using the patterns obtained from the mining process.**

Keywords: **Frequent Itemset Mining, Association Rule Mining, Big Data, Mapreduce.**

I. INTRODUCTION

Now a days social networking plays a major role and used as a communication media to share our thoughts to others. Big data mining and KDD (Knowledge Discovery in Database) are important techniques to discover hidden information from large datasets. Discovery of association rules from large database is one of the problems in KDD. Size and complexity of Big Data are challenges for discovering frequent itemset from the large datasets using frequent itemset mining. Association rule mining and frequent itemset mining is popular techniques of data mining. It reveals frequency of items purchased together. The whole database scan is necessary in FIM, it might create challenge when datasets size is scaling, as large datasets does not fit into memory. Several approaches exist for association rule mining. Frequent itemsets play an essential role in finding correlations, clusters, episodes and many other data mining tasks. Value discovered from frequent itemsets can be used to take decisions in competitive business environment. The main purpose of FIM techniques is to extract frequent itemsets from large databases. Agrawal et al. put forward Apriori algorithm which extracts frequent itemsets from the databases which having frequency greater than minimum support given. Enormous amount of work has been put forward to extract frequent items. There exist various parallel and distributed algorithms for processing big data but having memory and I/O cost limitations. MapReduce uses parallel computing approach and HDFS is fault tolerant system. MapReduce has two functions namely Map and Reduce functions. In this paper, based on BigFIM algorithm, a new algorithm optimizes the speed of BigFIM algorithm. At first by using parallel K-Means clustering, ‘n’ number of clusters are generated from large datasets. Then obtained clusters are mined using ClustBigFIM algorithm, effectively increasing the execution efficiency.

This paper is correlated as follows: Section II gives the survey of different frequent itemset mining algorithms and overview of existing system. Section III explains proposed model. Section IV explains about the implementation and result. Section V gives conclusion of the paper including future work.

II. LITERATURE REVIEW

Sandy Moens et al. use two methods for frequent itemset mining for Big Data on MapReduce, First method Dist-Eclat is distributed version of pure Eclat method which optimizes speed by distributing the search space evenly among mappers, second method BigFIM uses both Apriori based method and Eclat with projected databases that fit in memory for extracting frequent itemsets.

Agarwal et al. uses apriori algorithm for generating frequent itemsets and association rules. Level wise search, Monotonicity property was implemented in this paper. Frequency of an itemsets are counted by scanning the database D and then candidate k+1 itemsets are generated from frequent k-itemset by applying support and threshold condition.

Zaki et al. uses eclat algorithm which requires vertical database D needs to be stored in main memory. dEclat developed by Zaki and Gouda store diffset; diffset is difference between candidate itemset of size k and prefix frequent itemsets of size k-1; support value is based on diffset gaining performance growth than Eclat; it is not efficient when the database is sparse.

Malek and Kadima proposed a novel approach for discovering frequent itemsets from set of clusters of dataset using MapReduce which increases the performance. In map phase distance from centers are computed for each itemset and assigned to related cluster. In reduce phase partial sum of distances is calculated and new itemsets list is computed.

III. PROPOSED MODEL

The big data is stored in a hadoop distributed file system (HDFS) as blocks. Firstly by using parallel K-means clustering k number of clusters are created from the datasets. Then obtained clusters are mined using ClustBigFIM algorithm, to find the frequent itemsets. ClustBigFIM uses hybrid approach, clustering using k-means algorithm to generate Clusters from large datasets and Apriori and Eclat to extract frequent itemsets from generated clusters using MapReduce programming model. Then by using the frequent itemsets, patterns and rules are generated which is very much useful for the business purposes.

A. System Architecture:

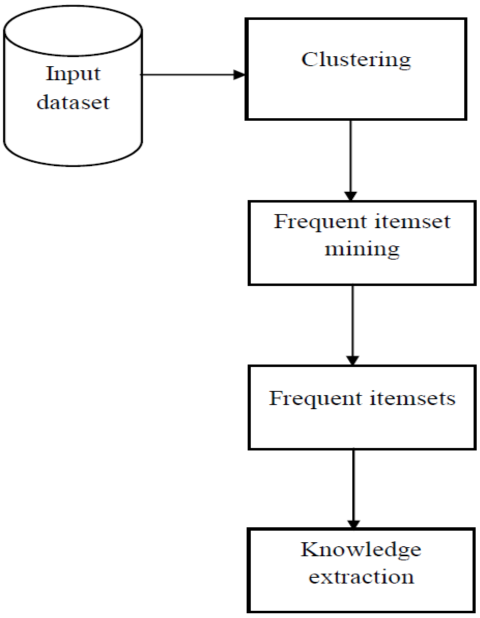

Figure 1: System architecture

The figure 1 shows the system architecture diagram of the proposed system. The input dataset is clustered by using k-means clustering. Several numbers of clusters are formed, then frequent itemsets are mined from the formed clusters using apriori algorithm. By using the frequent itemsets, knowledge extraction process is done which is very much useful for the business environment.

IV. IMPLEMENTATION AND RESULTS

The following are the modules present in the project.

- Finding clusters
- Finding k-FIs
- Generating single global TID list
- Mining frequent patterns

A. Finding Clusters:

K-means clustering algorithm is used for finding clusters from large datasets. Proposed algorithm implements parallel k-means algorithm for generating clusters using Compute_Dist function and combiner function which takes centers as input. Generated clusters are mined using Apriori algorithm in next step.

Clusters of transactions are formed based on below formula which calculates minimum squared error, and assign each transaction to the cluster. Input to this phase is transaction dataset and cluster size, clusters are generated like C={t1,t11,...t300}.
Input: Cluster Size and Dataset

Output: Clusters with size z

Steps:

1. Calculate distance between centers and transaction id in map phase.

2. Use combiner function to combine results of above step.

3. Compute MSE and assign similar points to clusters in reduce phase.

4. Repeat steps 1-3 by changing Centre and stop when convergence criteria is reached.

B. Finding K-FIs:

Frequent itemsets mining is done from clusters using Apriori algorithm. Local supports are searched using mappers and then global support is calculated by reducers. Apriori is used up to certain length to find frequent k-length prefixes, P_k= $\{p_2, p_3, \ldots, p_l\}$ like $P_2 = \{p_2, p_3, p_4, p_5, p_6\}$, Let two itemsets be X, Y $\subseteq$ I then monotonic property of support is, $X \subseteq Y \rightarrow$ support(Y) $\leq$ support(X). This property is used while pruning the itemsets from candidate list in order to obtain next frequent itemset list.

Transaction ID list for big datasets could not be managed by Eclat algorithm, So frequent itemsets of size k are mined from generated clusters in previous phase on the basis of minimum support condition by using apriori algorithm which handles problem of large datasets.

Input: size of cluster (s), Minimum threshold value (σ), prefix length (l)

Output: Prefixes with length l and k-FIs

Steps:

1. Support of all items in a cluster was calculated using Apriori algorithm.

2. Apply Support (xi)> σ and calculate FIs using monotonic property.

3. Repeat till calculating all k-FIs using mapper and reducers.

4. Repeat for all clusters and find final k-FIs.

C. Generating Single Global Tid List:

From computed prefixes in above step, prefix tree is built; tid_lists for (k+1) FIs are obtained which can be done similar to word counting. However supports and reducers compute single global tid_lists. Prefix tree are formed in a way that siblings are arranged by their individual frequency in increasing order. Formally, X = {i_1, i_2, ..., i_k}, where support(i_a) ≤ support(i_b) ↔ a < b which can be used for pruning. Mapper computes local–tid_lists instead to local support. The global TID list is generated by combining all local TID list using mappers and reducers.

Input: Prefix Tree, Min Supports

Output: Single TID list of all items

Steps:

1. Calculate TID list using prefix tree in map phase.
2. Create single TID list from TID list generated in above step.
3. Based on the condition support (i_a) ≤ support (i_b) ↔ a < b perform pruning operation.
4. Generate prefix groups, $P_k = (P_k^1, P_k^2, \ldots, P_k)$

D. Mining Frequent Patterns:

Next (k+1) frequent itemsets are extracted using Eclat algorithm. Prefix tree created in phase 2 is mined independently by mappers and frequent itemsets are generated.

Input: Generated prefix tree, Minimum support (σ).

Output: k-FIs

Steps:

1. Apply Eclat algorithm and find FIs till size k.
2. Repeat step 1 for each Subtree in map phase.
3. Find all frequent items of size k and store them in compressed format.

V. CONCLUSION

The ClustBigFIM algorithm has hybrid method for finding frequent item sets using parallel k-means, Apriori and Eclat algorithm on MapReduce framework. Parallel k-means can give approximate results but in short time; Apriori finds frequent itemsets having size k; Eclat algorithm finds potential extensions to frequent item sets and subtree mining by resolving memory problem. MapReduce platform can be used extensively for mining Big Data from social media as tradition tool and techniques cannot handle Big Data. Planning to apply frequent item set mining algorithm and MapReduce framework on stream of data which can be real time insights in Big Data. The FIM algorithm thus used in this project is based on MapReduce programming model. Pre-processing can be done by using K-means clustering algorithm, frequent itemsets of size k are extracted using Apriori algorithm and discovered frequent itemsets are mined using Eclat algorithm. In future the frequent itemsets can be discovered by using FP-Growth algorithm.

REFERENCES

[1] Moens, S.; Aksehirli, E.; Goethals, B., "Frequent Itemset Mining for Big Data," Big Data, 2013 IEEE International Conference on , vol., no., pp.111,118, 6-9 Oct. 2013 doi: 10.1109/BigData. 2013.6691742.

[2] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In Proc. VLDB, pages 487–499, 1994.

[3] M. J. Zaki and K. Gouda. Fast vertical mining using diffsets. In Proc. ACM SIGKDD, pages 326–335, 2003.

[4] R. Agrawal and J. C. Shafer. Parallel mining of association rules. Ieee Trans. On Knowledge And Data Engineering, 8:962–969, 1996.

[5] M. Malek and H. Kadima. Searching frequent itemsets by clustering data: towards a parallel approach using mapreduce. In Proc. WISE 2011 and 2012 Workshops, pages 251–258.

[6] T. White, Hadoop: The Definitive Guide. O'Reilly Media, Yahoo! Press, June 5, 2009.

[7] M. Zaki, S. Parthasarathy, M. Ogihara, and W. Li. Parallel algorithms for discovery of association rules. Data Min. and Knowl. Disc., pages 343–373, 1997.

[8] Zhigang Zhang; Genlin Ji; Mengmeng Tang, "MREclat: An Algorithm for Parallel Mining Frequent Itemsets," Advanced Cloud and Big Data (CBD), 2013 Int. Conf. on, vol., no., pp.177,180, 13-15 Dec. 2013 doi: 10.1109/ CBD.2013.22.

[9] L. Zeng, L. Li, L. Duan, K. Lu, Z. Shi, M. Wang, W. Wu, and P. Luo. Distributed data mining: a survey. Information Technology and Management, pages 403–409, 2012.