# Apply Apriori on Big Data for Frequent items mining

**Prabhakar Marry,** Asst.Prof. , Vignan Institute of Technology & Science, marryprabhakar@gmail.com

**Janardhan G,** Asst.Prof. , Vignan Institute of Technology & Science, janardhanyadav5@gmail.com

**Abstract:** For getting required information, one of the major task in pattern mining from un processed data . Finding regularity, sequence and homogeneity in data is aim of the task .many methods are proposed, these methods performance are draped in data growth in size .The apriori property is used for pruning computation space, its improve algorithm performance. So till now no algorithm proposed on big data using apriori, in this paper we proposed a new algorithm which use apriori property on bigdata ,which reduce computation space

Keywords: Apriori property, bigdata, frequent items

## Introduction:

Now a days for business decision making, the data has to convert raw into use full and meaningful .the technique involve in this purpose are deal with huge amount in value, unstructured and verity of data called big data[1].

In data mining and data analysis pattern mining [2] is important part, pattern mining is find the substructure and subsequences[3][4].this problem proposes in context of market basket problem finding frequent items[5] bought together[6],in [7,8,9,10] algorithm are based on apriori based.

In numerous application areas [12],find items set have in high number of transaction rather than existing all item sets , new method proposed based on anti-monotone property as a pruning strategy,anti-monotone property is super set is infrequent its all sub sets also infrequent ,so no need to compute the all subsets of superset. the traditional methods are not suitable for big data[13] presenting two main challenges to be solved: 1) computational complexity and 2) main memory requirements [14]. In this scenario, sequential pat-tern mining algorithms on a single machine may not handle the whole procedure and an adaptation of them to emerging technologies [15] might be fundamental to complete process

Pattern P is defined as $\{P = \{ij, \ldots, ik\} \subseteq I, j \geq 1, k \leq n\}$. Given a pattern P, its length or size is denoted as $|P|$, i.e., the number of singletons that it includes. Thus, for $P = \{i1, i2, \ldots, ik\} \subseteq I$, its size is defined as $|P| = k$. Additionally, given a set of all transactions $T = \{t1, t2, \ldots, tm\}$ in a dataset, the support of a pattern P is defined as the number of transactions that P satisfies, i.e., $supp(P) = |\{\forall tl \in T : P \subseteq tl\}|$. A pattern P is considered as frequent iff $supp(P) \geq thresh$.

If n is unique items ,N is no of transactions ,M=2n-1 frequent items to be generated ,comparison are $O(M \times N \times n)$ required in bute-force method. becomes computational complexity expensive for big data,for example, let us take a transactional dataset have 20 singletons and 1

million of records . Here by brute-force approach has to mine M = 220 − 1 = 1 048 575 item-sets and a 2.09 • 1013 comparisons to compute the support of each pattern

So computational pruning is required for big data, we proposed method which reduce computational complexity by apply on apriori on big data, rest of paper organized as in section 2 discussed about proposed method, experiment results are discussed in section3,section 4 conclude the our findings

**Proposed algorithm:**

The existing apriori algorithm (Algorithm 1) works as follows ,first find the item set in size with one ,eliminate the those are below the threshold value, next generate second item set by using first item set and apply threshold ,continue this process until no new item set are generated

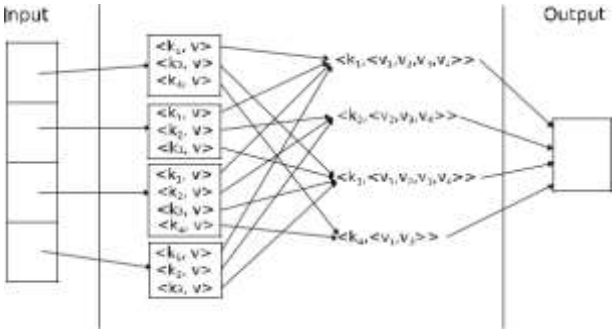



Fig 1: MapReducer framework

Work with big data we need follow map-reduce frame work ,its works as follows : input data is divided into parts for each parts generate key ,value pair ,reduce combine the all same key produce final value which shows in fig 1

Algorithm 1 Apriori Algorithm

Input: I // set of transactions

Output: O // list of patterns found in data

$L = \emptyset$

*for all* $r \in I$ *do*

*for* $(b = 1; b \leq |r|; b{+}{+})$ *do*

$B = \{\forall P : P = \{i_j, \ldots, i_n\} \wedge P \subseteq r \wedge |P| = b\}$

*// candidate item-sets in r*

*∀P ∈ B, then sup(P) = 1*

*if B ∩ L = ∅ then*

*∀P ∈ L : P ∈ B, then sup(P) + +*

*end if*

*L = L ∪{B \ L}  // include new patterns in L*

*end for*

*end for*

*return L*

we proposed new algorithm for computing frequent item sets on big data(Algorithm 2), Fig. 2 illustrates how the AprioriB algorithm works. In this example, the input database is divided into four sub-databases, and the AprioriMR algorithm includes four mappers and three reducers. As shown, each mapper mines item-sets (patterns) for its subdataset iterating transaction by transaction, producing a set of $P$, supp$(P)_l$ pairs for each transaction $t_l \in T$ . Then, in an internal MapReduce grouping proce-dure, $P$, supp$(P)_l$ pairs are grouped by the key $P$ producing $P$, supp$(P)_1$, supp$(P)_2$, . . . , supp$(P)_m$ pairs. Hence, taking the item-set $\{i_1 i_3\}$ as a key, the following pair is obtained$i_1 i_3\}$, 1, 1, 1, 1 . As described in Algorithm 2, the reducer phase is responsible for combining the values (supports for each item-set) and produce a final global support. Hence, taking the aforementioned item set $\{i_1 i_3\}$, the resulting value will be $i_1 i_3\}$, 4 . Finally, comprising $i_1$; the second reducer computes those item-sets including $i_2$; and, finally, $i_3$ is computed by the third reducer

Algorithm 2: AprioriB

begin procedure AprioriBig($t_l$)

*for (r = 1; r ≤ |$t_l$|; r++) do*

*B={ ∀P: P={$i_j$, ..., $i_n$} ∧ P ⊆$t_l$ ∧ |P|=r}*

*// candidate item-sets in tl*

*∀P ∈ B, then sup(P) = 1 // support is initialized*

*for all P ∈ B do*

*emit ( P, sup(P)$_l$ )// emit the key, value pair*

*end for*

*end for*

end procedure

begin procedure AprioriReducer( P, $sup(P)_1$, ..., $sup(P)_m$)

*s= 0*

*for all sup* $\in$ $sup(P)_1, sup(P)_2, ..., sup(P)_m$ *do*

*s += supp*

*end for*

*emit P, s*

end procedure

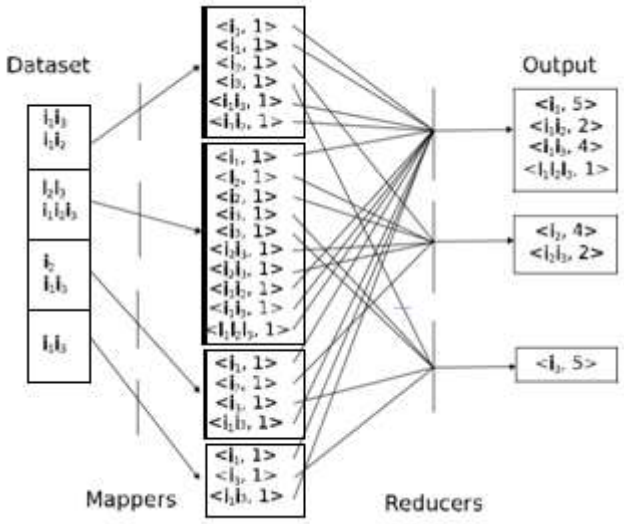
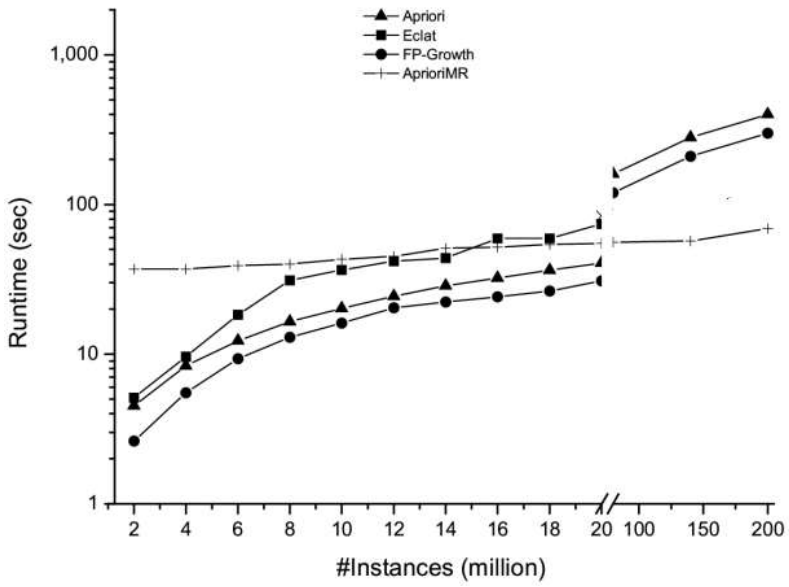



Fig2 :Diagram of AprioriB Algorithm

## Experiment:

For experiment we generate synthesized data set contain $2^3$ to $2^{20}$ instances the file size is 9.5 MB to 2.3 GB , Our algorithm performance study by different in volume Compare computational time both aprioriB and sequential pattern mining algorithms show in fig 3. We use Our experiments were run on HPC cluster comprising 12 compute nodes . Each o nodes two Intel Xeon E5645 CPUs with six cores at 2.4 GHz and 24-GB DDR memory. Cluster operating system was Rocks cluster 6.1 × 64. As for the specific details of the software Hadoop 2.6 used, with a maximum of 144 maps tasks, and a max of 3 reducers.




## Conclusion:

By using apriori on multiple map reduce, we generate all frequent item set with in available memory and reduce computational time, our algorithm aprioriB efficiently work for big data about 20 million to 200 million.. Results have also revealed the unsuitability of MapReduce frameworks when small datasets are considered.

## References:

[1] T.-M. Choi, H. K. Chan, and X. Yue, "Recent development in big data analytics for business operations and risk management," IEEE Trans. Cybern., vol. 47, no. 1, pp. 81–92, Jan. 2017. [Online]. Available: http://dx.doi.org/10.1109/TCYB.2015.2507599

[2] J. M. Luna, "Pattern mining: Current status and emerging topics," Progr. Artif. Intell., vol. 5, no. 3, pp. 165–170, 2016.

[3] C. C. Aggarwal and J. Han, Frequent Pattern Mining, 1st ed. Cham, Switzerland: Springer, 2014.

[4] J. Han, H. Cheng, D. Xin, and X. Yan, "Frequent pattern mining: Current status and future directions," Data Min. Knowl. Disc., vol. 15, no. 1,pp. 55–86, 2007.

[5] J. M. Luna, J. R. Romero, C. Romero, and S. Ventura, "On the use of genetic programming for mining compre-hensible rules in subgroup discovery," IEEE Trans. Cybern., vol. 44, no. 12, pp. 2329–2341, Dec. 2014. [Online]. Available: http://dx.doi.org/10.1109/TCYB.2014.2306819

[6] R. Agrawal, T. Imielinski, and A. Swami, "Database mining: A performance perspective," IEEE Trans. Knowl. Data Eng., vol. 5, no. 6,pp. 914–925, Dec. 1993.

[7] J. Han, J. Pei, Y. Yin, and R. Mao, "Mining frequent patterns with-out candidate generation: A frequent-pattern tree approach," Data Min. Know. Disc., vol. 8, no. 1, pp. 53–87, 2004.

[8] S. Zhang, Z. Du, and J. T. L. Wang, "New techniques for min-ing frequent patterns in unordered trees," IEEE Trans. Cybern., vol. 45, no. 6, pp. 1113–1125, Jun. 2015. [Online]. Available: http://dx.doi.org/10.1109/TCYB.2014.2345579

[9] J. M. Luna, J. R. Romero, and S. Ventura, "Design and behavior study of a grammar-guided genetic programming algorithm for min-ing association rules," Knowl. Inf. Syst., vol. 32, no. 1, pp. 53–76, 2012.

[10] R. Agrawal, T. Imielinski,´ and A. Swami, "Mining association rules between sets of items in large databases," in Proc. ACM SIGMOD Int. Conf. Manag. Data (SIGMOD), Washington, DC, USA, 1993,pp. 207–216.

[11] J. Liu, K. Wang, and B. C. M. Fung, "Mining high utility patterns in one phase without generating candidates," IEEE Trans. Knowl. Data Eng., vol. 28, no. 5, pp. 1245–1257, May 2016. [Online]. Available: http://dx.doi.org/10.1109/TKDE.2015.2510012

[12] S. Ventura and J. M. Luna, Pattern Mining With Evolutionary Algorithms, 1st ed. Cham, Switzerland: Springer, 2016.

[13] S. Moens, E. Aksehirli, and B. Goethals, "Frequent itemset mining for big data," in Proc. IEEE Int. Conf. Big Data (IEEEBigData), Silicon Valley, CA, USA, 2013, pp. 111–118.

[14] J. M. Luna, A. Cano, M. Pechenizkiy, and S. Ventura, "Speeding-up association rule mining with inverted index compression," IEEE Trans. Cybern., vol. 46, no. 12, pp. 3059–3072, Dec. 2016. [Online]. Available: http://dx.doi.org/10.1109/TCYB.2015.2496175

[15] S. Sakr, A. Liu, D. M. Batista, and M. Alomari, "A survey of large scale data management approaches in cloud environments," IEEE Commun. Surveys Tuts., vol. 13, no. 3, pp. 311–336, 3rd Quart., 2011.

Authors Profile :

**Prabhakar Marry** M.Tech(CS) from RRS College of Engineering, JNTUH and B.Tech(IT) from Swami Ramananda Tirtha Institute of Science & Technology. He is having more than 10 years of experience has guided UG & PG students, currently he is working as Asst.Prof. at VIGNAN INSTITUTE OF TECHNOLOGY & SCIENCE. His research areas include Data Mining, Software Engineering, Computer Networks, Cloud computing, Design Analysis of Algorithms.

**Janardhan G** M.Tech(CSE) from NITS College of Engineering, JNTUH and B.Tech(CSE) from Swami Ramananda Tirtha Institute of Science & Technology. He is having more than 7 years of experience has guided UG & PG students, currently he is working as Asst.Prof. at VIGNAN INSTITUTE OF TECHNOLOGY & SCIENCE. His research areas include Data Mining, Database Management System.