International Research Journal of Computer Science (IRJCS)
Issue 5, Volume 2 (May 2015)
ISSN: 2393-9842
www.irjcs.com

Parallel Implementation of Hyperclique Miner Algorithm for Association Analysis of Weighted Protein-Protein Interaction Network

Sarbani Dasgupta*
Department of MCA
Techno India College of Technology

Banani Saha
Department of Computer Science and Engineering
University of Calcutta

Abstract -Association analysis is an important aspect of data mining which is used for discovering important relationship among the data stored in large databases. In bioinformatics, association analysis is used for analyzing associations among the genes and proteins. Protein interaction network is an important part of bioinformatics that depicts the physical interactions among the proteins. It is used for predicting functions of the proteins. Association analysis of the protein interaction network provides information regarding frequently occurring proteins. Several algorithms have been proposed for association analysis of the protein-protein interaction network. Among them, the hypercliqueminer algorithm is used for elimination of noise from the protein-protein interaction network. It also addresses the problem of incompleteness in the protein-protein interaction network. This algorithm was implemented sequentially. But with the increase in size of the database, this algorithm will consume a lot of processing time. To address this problem, a parallel version of Hypercliqueminer algorithm has been proposed in this paper. The algorithm has also been implemented on Hadoop platform.

Keywords: Association analysis, Protein-protein interaction network, Hyperclique miner algorithm,H-confidence, Hadoop.

I. INTRODUCTION

Data mining is an analytical process of extracting knowledge from already existing large databases. Association rule mining (ARM) [1] is an important technique of data mining which is used for finding correlation among the data in the large databases. ARM was traditionally used for market basket analysis of the transactional databases [2].The main aim of the algorithm is to find frequent itemsets among the items present in the database. Several algorithms namely the classical Apriori algorithm [2], Éclat algorithm [4], FP-growth algorithm [3] have been proposed for ARM from the large databases. ARM algorithm can be applied in several fields like webmining, social networking and bioinformatics.

Bioinformatics [5] is the process of application of computer technology for management of biological data. Protein is an important macromolecule of cells. Protein is responsible for several cellular functions. Protein never performs function in isolation. It always interacts with other proteins to perform its function. In bioinformatics, protein-protein interaction (PPI) network [5] depicts the physical interactions among the proteins. It is used for predicting functions of proteins. PPI network [6] is represented by an undirected graph consisting of nodes and edges. In the graph, each node denotes a protein and the edges denote the pair wise interaction among the proteins. Several approaches have been proposed for predicting functions of proteins from PPI network. Among them, association analysis based approach [6] is used for identification of frequently occurring protein pairs in protein interaction network. The sub graphs obtained from the frequent analysis of the protein interaction network denote a functional module which is used for determining the function of the protein.

The PPI network has two disadvantages. The main disadvantage is the presence of noise [6] in form of spurious edges in the network, analysis of which may give inappropriate result. The second disadvantage is the incompleteness of the PPI network [6] i.e the biologically valid interaction among the proteins are absent in the PPI network. To overcome the above-mention disadvantages, the PPI network is converted into weighted graph. The weights are assigned to the edge of the PPI network which denotes the strength and reliability of the interaction between the proteins depicted by nodes of the graph.

The association analysis based approach uses hyperclique miner [7] algorithm for analysis of weighted PPI network. The algorithm has been implemented sequentially which consumes a lot of processing time. In this paper, a parallel implementation of the hyperclique miner algorithm has been proposed to overcome the problems of sequential algorithm.

The rest of the paper is organized as follows: section II discusses about the existing work. The proposed approach has been detailed out in Section III. The advantages of the proposed approach over the existing sequential implementation approach are discussed in Section IV. The article is concluded in the last section.

II. EXISTING APPROACHES

Protein is a biological element which is responsible for many important function of an organism. The protein cannot perform any function on its own. It usually interacts with several other proteins to accomplish its task. There are two types of interactions that occur among the proteins, mainly genetic interaction and physical interaction. Physical interaction defines the process through which protein accomplishes its function [5]. This physical interaction among the proteins is depicted in a network called PPI network. It is usually represented as an undirected graph where the nodes denote the protein and the edges denote the interaction among them.

Several approaches have been proposed for determining the functions of the proteins [5] namely neighborhood based approach [8], global optimization based approach [9], clustering based approach [10] and association analysis based approach [11].

As stated earlier, association analysis of unweighted PPI network has several disadvantages. The hyperclique pattern of ARM algorithm, which is used for removal of noise from the binary data [12], has been applied for the purpose of noise removal from the PPI network [11]. Hyperclique pattern are those patterns that are strongly correlated with each other. Unlike frequent pattern which use support and confidence as the measurement for finding frequent pattern, hyperclique pattern use h-confidence (hconf)[5] which reflects the overall affinity among the items in a pattern. The hconf is defined as

$$\text{hconf}(X) = \text{sup}(i_1, i_2, \ldots, i_k)/\max[\text{sup}(i_1), \text{sup}(i_2), \ldots, \text{sup}(i_k)), \quad (1)$$

where $\text{sup}(i_j)$ is the support of an item i_j in an itemset.

The data items which satisfies hconf $\geq \delta$, (where δ is specified by the user) is a hyperclique pattern.
The hconf can be used to measure common neighborhood similarity between two proteins.[6] in a PPI network.
For this purpose the hconf is defined as

$$\text{hconf}(p1,p2) = \min\left(\frac{x}{Np1}, \frac{x}{Np2}\right), \quad (2)$$

where $x = N_{p1} \cap N_{p2}$, N_{p1} and N_{p2} denote of neighboring protein of proteins p1 and p2 respectively.

For a weighted graph[6], x,N_{p1} and N_{p2} are replaced by the following values

x= sum of minimum of weights of each pair of edges that are incident on a protein p from both p1 and p2.

N_{p1}= sum of weights of edges incident on p1.

N_{p2}=sum of weights of edges incident on p2.

The hconf value is guaranteed to fall within the range of 0 and1 [12].

From this definition it has been hypothesized that protein pairs having a high h − confidence score are expected to have a valid interaction between them, and the interactions between protein pairs having a low h − confidence score are expected to be noisy. Based on the above value of h-confidence Pandey et al[5] has proposed a graph transform approach for the protein interaction dataset. In this approach at first the hconf measure is computed between each pair of protein. After this a user specified threshold is applied to eliminate protein pairs having hconf less than the user specified threshold. The resultant graph obtained after the elimination of the spurious edge is expected to be less noisy and eliminate incomplete edges and contain more accurate weight on the edges. Though the disadvantages of the PPI network can be eliminated by assigning weights to the edges of the graph and application of the hyperclique miner algorithm, still there exist some problem because the algorithm is implemented sequentially which may consume a lot of processing time. The following section discusses about the parallel implementation of the Hyperclique miner algorithm.

III. PROPOSED APPROACH

The existing approach as discussed above was implemented on a single machine. However the sequential implementation of the algorithm will consume a lot of processing time. The hyperclique miner algorithm, used for association analysis of the PPI network is implemented parallelly .

The parallel implementation is done by Apache Hadoop [13], an open-source software framework. In Apache Hadoop, data can be stored and processed in a distributed environment across clusters of computers using simple programming model.

A. Hadoop software

Hadoop[13] is an open-source software framework used for storing and processing data in a distributed fashion on large clusters of commodity hardware. Essentially, it accomplishes two tasks: massive data storage and faster processing. The Hadoop software framework has three important part namely MapReduce programming model[14] and HDFS[14] and Yarn[14].

1. MapReduce programming model

Google's Map Reduce paradigm [15], [16], [17] is used by the Hadoop software for processing of data. This programming model is used for processing large datasets distributed across several machines in a cluster. MapReduce programming model hides issues of distributed and parallel programming i.e. load balancing, performance of network, fault tolerance and locality optimization [16] from the programmer.

Map () function accepts the key/value pair of the input dataset and produces as output list of intermediate key/value pair. It can be expressed as

$$\text{Map (inputkey, input value)} \rightarrow \text{list (outputkey, intermediatevalue)} \quad (3)$$

The Reduce () function will accept the output key and the list of values for the particular key. It will combine the list of values to form a smaller output list. The Reduce () function is expressed as following:

$$\text{Reduce (output key, list(intermediate value))} \rightarrow \text{list (outputvalue)} \quad (4)$$

2. Hadoop distributed File System (HDFS)

The concept of Hadoop distributed File System (HDFS)[14] is a distributed file system that can execute on commodity hardware. It can be used to store and process large datasets distributed across clusters of computers. HDFS follows master/slave architecture [16]. The HDFS cluster has one master server known as Name Node [14]and several Datanodes[14]. The Namenode records the operation within the file system. The Datanode manages the storage of the machine where it is executed. In HDFS the user input file is divided into several blocks which are stored in a set of DataNodes. The Namenode decides the allocation of the blocks to a particular DataNode. The DataNodes are responsible for executing read and write instruction of the file system's clients. The NameNode instructs the DataNodes to create, delete, and replicate data blocks.

3. YARN

YARN [14] is a resource management framework for scheduling and handling resource requests from distributed applications.

B. Parallel Hypercliqueminer algorithm

Input:

1. A database β of m number of transactions $\alpha = \{\alpha_1, \alpha_2 \ldots, \alpha_M\}$, each $\alpha_i \in \alpha$consists of attributes $\{j_1, j_2, \ldots. j_k\}$.
2. Minimum h-confidence value ($minh_c$).
3. Minimum support count (sup)

Output: Hyperclique patterns.

Procedure
{
Scan the database β
Generate support count of every item in the database using Map_itemset_1 ()
and Reduce_itemset_1 () function.
Generate the candidate set of size 1 denoted by CS_1 by removing items whose supportcount<sup and arranging the remaining item in alphabetic order.
For 1 < itemsize < K-1
do
{
Generate hyperclique pattern using Map () and Reduce ().
Generate potential hyperclique pattern by removing pattern whose h-confidence<$minh_c$.).
}

}
Procedure Map_itemset_1()
{The input database β is stored into HDFS.
Split it into different files represented as <key, value>.
Send it to the map () present at each node to generate an output represented as <key, value>.
}
Procedure Reduce_itemset_1()
{Input <key, value> obtained from the map function.
Add up the values with same key values ,the occurrence count of each item of the transaction.
Evaluate the support count by dividing the occurrence of the items by the number of transaction.
}
Procedure Map ()
{Self join candidate itemset of size 1to generate candidate itemset of size2.
Split it into different files represented as <key,value>.
Processed by the map() at each node and generate output represented as <key,value>
}
Procedure Reduce ()
{Input <key, value> pair obtained from the map ().
Add up the values with same key values to get occurrence count of each itemset .
Obtain the support count after dividing the occurrences of the itemset by the number of transaction.
Elininate the itemset whose hconf<minhc
}

IV. PERFORMANCE ANALYSIS

There exist many PPI network for different species. These PPI network information are stored in several databases, like DIP database [],MIPS database. In this paper we have proposed an parallel implementation of the hyperclique miner algorithm for association analysis of weighted PPI network.The gain in performance of the parallel system due to implementation using Hadoop software is observed. The Hadoop is installed in multicores machine. It has been found that as the number of cores increases, there is a remarkable improvement in performance of the algorithm. The parallelization of the algorithm is carried out using 4 machines each consisting of multiple cores AMD processors. The gain in performance of any parallel algorithm is evaluated based on three properties namely scaleup[18], sizeup[18], speedup[18]. The scaleup factor determines the processing ability of a parallel system to compute a larger job at the same runtime as that of a single system.It is expressed as

$$\text{Scaleup}(\text{data}, x) = \frac{T_1}{T_{xx}}, \quad (5)$$

where T_1 = the execution time for processing data on 1 core
T_{xx} is the execution time for processing the n times of the data on a multi core.
The sizeup factor is defined as the ratio between time taken by the system to process n times larger data to the time required for processing the data.The sizeup factor is denoted as

$$\text{Sizeup (data, n)} = \frac{T_n}{T_1}, \quad (6)$$

where T_n =is the time required for processing n times larger data
T_1 = the time required for processing the data.
The speed of processing will increase with the increase in number of cores of processor, even if the size of the dataset increases. It is shown by the following formula,

$$Speedup = \frac{T_1}{T_m}, \quad (7)$$

where T_1 = the processing time of a single processor
T_m =the time required for processing data by m number of processors.

To show that the scalup properties we have increased the size of the dataset as well as the no.of cores of the processor. Table 1(a) lists the scalup factor and the number of cores of processors. It is shown with the help of graph in Figure 1 that as the number of cores of the processor increases there is a decrease in the scalup factor for the same dataset. To show the sizeup factor Table 1(b) is considered where the size of the dataset and the sizeup factors are tabulated. Fig 2 shows that the as the size of the dataset increases there is a increase in the sizeup factor which means that the algorithm will show better result with the increase

in dataset size. Table1(c) show that the speed of processing will increase with the increase in number of cores of processor, even if the size of the dataset increases. The improvement in the performance of the algorithm with time is depicted in Fig.3.The graph shows that as the number of cores in the processor increases, the speed of processing is also increases, resulting faster analysis of the Hyperclique miner algorithm.

No.of cores	ScaleUp
4	1
8	.87
16	.79
32	.69
48	.61

Datasetsize	Sizeup
1GB	1.2
2GB	2.7
4GB	3.8
8GB	4.5

No.of cores	Speed Up
4cores	0.8
8cores	6
16 cores	7
32 cores	8.5
48 cores	9.8

Table1. Different types of evaluation factors (a) ScaleUp factors in case of dataset size 8GB.Processing (b)SizeUp factors (c)Speed in case of large dataset of size 8GB.

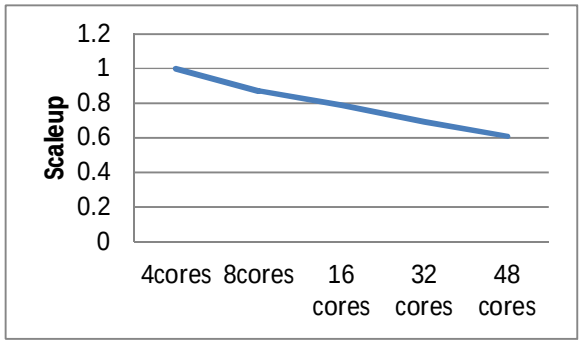

Fig1: Scale up factor of the parallel algorithm using dataset of 8GB

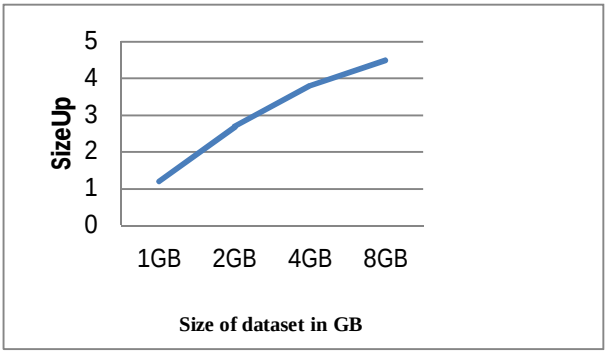

Fig 2: SizeUp of the parallel algorithm Vs Size of dataset

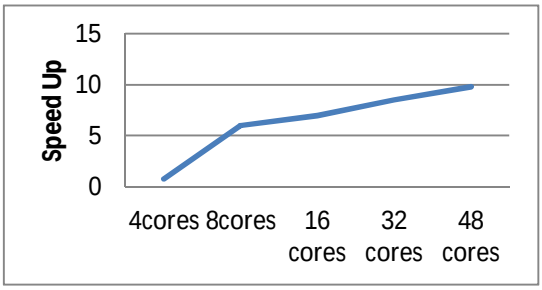

Fig3: Runtime of the parallel algorithm using dataset of 8GB

From the above graphs it can be shown that the parallel implementation of the Hyperclique miner algorithm will compute faster than the sequential algorithm. Therefore it can be said that the proposed algorithm will efficiently mine weighted PPI network .

V. CONCLUSION

Association analysis is applied field of bioinformatics it can for analysis of the PPI network . Hyper clique miner algorithm is used for association analysis of PPI interaction network. In this paper we have parallely implemented the traditional hyper clique miner algorithm used for construction of hyper graph useful for the protein-protein interaction network. In conclusion, significant scope exists for future research on designing novel association analysis techniques for complex biological data sets and their associated problems. Such techniques will significantly aid in realizing the potential of association analysis for discovering novel knowledge from these data sets and solve important bioinformatics problems.

REFERENCES

[1] Agrawal, R., Imielinski, T., Swami, A.N.: Mining association rules between sets of items in large databases. In: Proc. SIGMOD, pp. 207–216 (1993).
[2] Agrawal, R. and R. Srikant: 1994, `Fast algorithms for Mining association rules'. In: *Proc. of the 20th International Conference on Very Large Data Bases*.
[3] Han, J., Pei, J., Yin, Y., Mao, R.: Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach. Data Mining and Knowledge Discovery 8(1), 53–87 (2004).
[4] Christian Borgelt , Efficient Implementations of Apriori and Eclat. Workshop of Frequent Item Set Mining Implementations (FIMI 2003, Melbourne, FL, USA).
[5] Pandey, G., Kumar, V., Steinbach, M.: Computational approaches for protein function prediction: A survey. Technical Report 06-028, Department of Computer Science and Engineering, University of Minnesota (October 2006).
[6] G.Atluri, R.Gupta, Gang Fang, Pandey,M. Steinbach, V. Kumar.:' Association Analysis Techniques for Bioinformatics Problems', S. Rajasekaran (Ed.): BICoB 2009, LNBI 5462, pp. 1–13, 2009. c_Springer-Verlag Berlin Heidelberg 2009.
[7] Xiong, H., Tan, P.-N., Kumar, V.: Hyperclique pattern discovery. Data Min. Knowl. Discov. 13(2), 219–242 (2006).
[8] C. Lin, D. Jiang, and A. Zhang. Prediction of protein function using common-neighbors in protein-protein interaction networks. In *Proc. IEEE Symposium on BionInformatics and BioEngineering (BIBE)*,pages 251–260, 2006.
[9] J. B. Pereira-Leal, A. J. Enright, and C. A. Ouzounis. Detection of functional modules from protein interaction networks. *Proteins*,54(1):49 57, 2003.
[10] E. Nabieva, K. Jim, A. Agarwal, B. Chazelle, and M. Singh.Whole-proteome prediction of protein function via graph-theoretic analysis of interaction maps. *Bioinformatics*, 21(Suppl. 1):i1–i9, 2005.
[11] H. Xiong, X. He, C. Ding, Y. Zhang, V. Kumar, and S. R. Holbrook.Identification of functional modules in protein complexes via hyperclique pattern discovery. In *Proc. Pacific Symposium on Biocomputing (PSB)*, pages 221–232, 2005.
[12] H. Xiong, G. Pandey, M. Steinbach, and V. Kumar. Enhancing data analysis with noise removal. *IEEE Transactions on Knowledge and Data Engineering*, 18(3):304–319, 2006.
[13] White Tom, "Hadoop :The Definitive Guide", O'reilly,3rd edition ISBN: 978-1-449-31152-0.
[14] "HDFS High Availability Using the Quorum Journal Manager." Apache Software Foundation. Available at http://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/HDFSHighAvailabilityWithQJM.html. Accessed on June 5, 2013.
[15] R. Agarwal and J. Shafer, "Parallel mining association rules", IEEE Trans. On Knowledge and Data Engg., 8(6):962-969, December 1996, pp. 4-6, 14.
[16] M. J. Zaki, S. Parthasarathy and W. Li., "Parallel data mining for association rules on shared memory multi-processors". In Supercomputing 96, Pittsburg, PA, November 1996, pp. 17-22.
[17]M. J. Zaki, S. Parthasarathy, M. Ogihara and W. Li, "New algorithms for fast discovery of association rules", in Proc. of 3rd Int'l. Conference on Knowledge Discovery and Data Mining, August 1997, pp. 283-296.
[18] NingLi ,Li Zeng,Qing He and Zhongzhi Shi,"Parallel Implementation of Apriori algorithm based on MapReduce",in Proc. of 13th ACIS International Conference on Software Engineering,Artificial Intelligence, NetworkingandParallel/Distributed Computing,@IEEE2012.

