An Improved PrePost Algorithm for Frequent Pattern Mining with Hadoop on Cloud

Sanket Thakare\textsuperscript{a}, Sheetal Rathi\textsuperscript{b}, R.R.Sedamkar\textsuperscript{c}

\textsuperscript{a}ME Scholar, Thakur College of Engineering and Technology, Mumbai-400101, India
\textsuperscript{b}Assistant Professor, Thakur College of Engineering and Technology, Mumbai-400101, India
\textsuperscript{c}Professor, Thakur College of Engineering and Technology, Mumbai-400101, India

Abstract

Due to the advancement in internet technologies the volume of data is tremendously increasing day by day. The research is gaining importance in extracting valuable information from such huge amount of data. Many research works are done and various algorithms are proposed. The PrePost algorithm is one of well-known algorithms of frequent pattern mining. It is based on N-list data structure to mine frequent item-sets. But the performance of PrePost algorithm degrades when it comes to processing of large amount of data. Hadoop is very well known technique for processing such large amount of data. This paper proposes the Improved PrePost algorithm which combines the features of Hadoop in order to process large data efficiently. Efficiency of PrePost algorithm is enhanced by implementing compact PPC tree with the general tree method and finding frequent itemsets without generating candidate itemsets. An architecture of the Improved PrePost algorithm with public cloud is proposed. The results show that as dataset size is increased, the Improved PrePost algorithm gives 60% better performance.

Keywords: Big Data, Data Mining, PPC Tree, Data Node, Name Node, Cloud Computing, S3 Storage

1. Introduction

Data mining is a computational process of discovering patterns in large datasets [1]. It is becoming the hot topic for researcher now a day because of its wide use in different area. Frequent pattern mining is a one of the most important and popular topics which plays an essential role in many data mining tasks. For example, association rule mining, sequential pattern mining, correlation mining, structured pattern mining and so on. As Big Data rises day by day, mine such huge amount of data is becoming a challenging task. The PrePost is a data mining algorithm for finding frequent itemsets [2]. It uses N-list data structure to represent itemsets. All the required information of itemsets is to be stored by N-list. Efficiency of PrePost algorithm is achieved by using the method of generating frequent itemsets without generation of candidate itemsets. It needs to scan database twice to construct PPC tree. Then, it generates N-list with
the help of PPC tree. The PPC tree consists of Null as a root and set of item prefix subtree as children. Each node in a subtree consists of item-name, count, children-list, pre-order and post-order. The item-name represents node name. The count represents the number of transactions presented by item. The children-list represents all the children of the node. Pre-order represent Pre-order rank of a node. Post-order represents post-order rank of a node. Each node in PPC tree has structure like <Pre-order, Post-order: count>. The PrePost algorithm uses PrePost code to generate FIM1 N-list. In the next step, it merges FIM1 N-lists and generates a final N-list. The Complexity of the algorithm is O \((m + n)\) where m and n are the length of two N-lists. The PrePost+ is an extension of PrePost algorithm proposed by Zhi-Hong Deng, Sheng-Long Lv for finding frequent itemset [3]. It uses an N-list to represent itemsets and discover frequent itemsets with a set-enumeration search tree. It also uses an efficient Children–Parent Equivalence pruning strategy which greatly reduces the search space. PrePost+ gives better performance as compared to FP-growth, but it consumes more memory. These algorithms give better performance when dealing with small datasets. But, the performance of this algorithm start decreases as soon as dataset size increases.

In this paper, proposed the PrePost algorithm with Hadoop in a distributed environment to deal with such large amount of datasets. MapReduce technique provides a distributed environment for execution and makes the PrePost algorithm capable of handling large datasets. Generally, physical cluster of Hadoop is built on homogeneous machines. Cloud services provide an easy way to setup Hadoop cluster with heterogeneous machines. Cloud services already offer different types of machine to the end user for different type of workloads. This is an easy solution to handle jobs which require variable resources. For example, with AWS cloud, Hadoop cluster can be launched with m2 type large machine if the Hadoop job requires more memory and with c1 type large machine if required more compute intensive margin. Now days, many organizations are using cloud servers for storing their business data. For the analysis of data, moving large volumes of data from cloud to the Hadoop cluster in organization premises is an inefficient and time consuming process. In this paper, also proposed an architecture of the PrePost algorithm on a public cloud. Running Hadoop on the cloud offers flexibility and reduces execution time. Instead of adding a node to Hadoop cluster, one can scale up or scale down machines in the Hadoop cluster as per need.

2. Related Work

Hadoop is an apache source framework based on Java that allows distributed processing of large datasets across a cluster of computers [4]. The Hadoop framework provides the environment of distributed storage and computation across clusters of computers. The main goal of Hadoop design is to scale up of a single server to thousands of machines, each offering local computation and storage. Hadoop uses the Google MapReduce technique to perform an operation in a Hadoop cluster [5]. Hadoop stores data in HDFS (Hadoop Distributed File System). The default block size of HDFS is 64MB or 128 MB. It is a Java based framework which is compatible with different platforms. This feature makes Hadoop efficient tool for Big Data. On the other hand, Cloud computing is also a very popular topic in the field of IT. It is an on demand IT resources and services via internet with a browser [6]. It also provides flexibility, scalability, elasticity, and reliability features. There are three types of cloud computing: public cloud, private cloud and hybrid cloud. Depending on the service models, cloud is classified as: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). Amazon provides the Amazon web services (AWS) [7] which is a large-scale distributed, transactional IT infrastructure. With the help of the AWS, large compute power, storage, and other services can be gotten in a minute and have the flexibility to choose the development platform or programming model. An EC2 instance is a virtual server that provides resizable computing capacity in the Amazon web services. The Amazon S3 is the web services to store and retrieve data at any point of time from anywhere with a browser. It require secret key and access key to communicate with s3 storage. It is a type of consistence storage. It is easy to use with a simple interface to store and retrieve the data. It also provide services like auto scaling and load balancing to handle dynamically increases load. The alarm service of AWS gives notification when load is increases on EC2 instance or any EC2 instance crashes.

3. Literature Survey

Frequent pattern mining was first introduced by the Agrawal [8] in 1991 for the market basket analysis. The goal of mining association rule is to detect and analyze the behavior of customer from association of different item brought
from the supermarket. The most famous example of an association rule is a customer who buys diapers also frequently
buys beers. Most of the research is done till 2000 in the algorithms like Apriori which is scanning database repeatedly
and very costly with long pattern [9]. In 2000, FP-Growth method is introduced by Jiawei Han which is based on
divide and conquer way [10]. FP-tree has better performance than Apriori algorithm. But, for mining frequent pattern
it has to create large amounts of conditional pattern and corresponding tree. Both the algorithms when faced large
dataset, its computational cost and execution time increases.

In 2012, DENG ZhiHong, WANG Zhong Hui & JIANG JiaJian proposed a new algorithm for fast mining
using N-lists called PrePost algorithm. The PrePost works on a data structure called N-list which stores all information
of itemsets. By avoiding scanning database repeatedly and mining without candidate generation, the PrePost achieves
the high efficiency as compared to others. But when the PrePost faces large dataset its execution time is increased.
The PrePost+ is the high performance, frequent pattern mining based on a set-enumeration search tree. It reduces the
search tree space with efficient Children–Parent Equivalence pruning strategy. But similar to PrePost algorithm, when
PrePost+ faces large dataset its execution time is also increased. In this paper, the PrePost algorithm implemented
with Hadoop framework. Now a days, Big data and cloud computing are both the fastest-moving technologies in the
IT industry identified in Gartner Inc.’s 2012 Hype Cycl. For the better performance of algorithm with large datasets,
the whole Hadoop cluster is implemented on the public cloud with Amazon web services. With the help of cloud, user
can use service anytime, anywhere with the browser. The cloud computing technology has large amount of computing
power with aggregating resources. It became powerful architecture to deliver solution for large scale datasets and
complex computing. The Efficiency of the PrePost algorithm can be increased with large scale data sets by
implementing Hadoop on cloud environment.

The next paper is organized as follows. Section 4 presents the implementation of the improved PrePost
algorithm with the Hadoop framework for finding frequent itemsets. Section 5 introduces implementation of the
improved PrePost Algorithm with Hadoop on Amazon web services Cloud environment. Section 6 presents
experimental results and discussion. Section 7 summarizes our study and some future research issues.

4. The Improved PrePost Algorithm With Hadoop

The PrePost algorithm is a data mining algorithm for frequent itemsets which uses N-list data structure to represent
the itemsets. All the required information of the itemsets is to be stored by N-list. Efficiency of the PrePost algorithm
is achieved by using the method of generating frequent itemsets without generation of candidate itemsets. The PrePost
algorithm is implemented with Hadoop to improve its performance. In the Improved PrePost algorithm, general tree
method is used to traverse the tree PPC tree. The general tree method used the same technique that is used by linked
based binary tree. Linked based binary tree uses linked list which is an implementation of the List interface. It provides
sequential access and more efficient for inserting and deleting items in the list. But, it became less efficient while
accessing items in the list. In the Improved PrePost, general tree method is implemented with Hash Map which is an
implementation of the Map interface. It provides an efficient for locating value based on the key. It does not store the
item in the order and it provides an easy way to access and delete items on the basis of key value pairs. The Improved
PrePost algorithm performs a mining operation in 5 stages [11]. The Fig. 1. shows the flow of the Improved PrePost
algorithm. It uses 5 Map Reduce function to find Nlist of frequent itemsets. The Improved PrePost uses distributed
cache function in the Hadoop framework for comparing two input files. The Distributed cache [12] will copy the
necessary files on to the slave node before executing any task on that node. It used to distribute simple, read-only text
files on each data node in the Hadoop cluster.

The main Steps of PrePost algorithm are as follows:

Step 1: The data file is given as input to the Hadoop. It divides whole input file into fixed size blocks called chunks,
and map it to the different DataNode in Hadoop cluster. DataNode counts the number of items in each block.
Then, apply support count and arrange all items in the descending order. Then, reducer combines data from
all DataNode and generate list called F1 list.

Step 2: The F1 list is mapped to different DataNode with the distributed cache. The main input file is rearranged
according to F1 list. Here, uses the concept of distributed cache to compare two files with Map. Then, generate
list called FIM1 list.
Step 3: The FIM1 list is passed as input and generate a compressed tree called PPC tree similar like FP tree.

Step 4: After generating PPC tree, uses post-order traversal to determine post-order and pre-order traversal to determine pre-order. The tree is not a type of binary tree. So, use general tree traversal method for finding pre-order and post-order.

Step 5: The post-order and pre-order rank of each node generate N-list for each node in the tree. Merge two N-list with condition <p.pre-order, p.post-order, q.count> where p.pre-order < q.pre-order and p.post-order > q.post-order.

As compare to the FPGrowth algorithm, the Improved PrePost algorithm is less complex and much more efficient. The FPGrowth algorithm uses FPTree in the whole process of mining, which makes the algorithm more complex [13] [14]. While the PPC tree in PrePost algorithm is used only for generating the PrePost code of each node. After generating the PrePost code, tree can be deleted.

5. The Proposed Architecture of Improved PrePost Algorithm on AWS Cloud

The proposed architecture is based on public cloud environment. The Improved PrePost Algorithm has been implemented with hadoop on cloud to evaluate its performance. Fig. 2. Shows the architecture of the Improved PrePost algorithm Amazon web services where hadoop as well as data is on AWS. In AWS, Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) [15]. It reduces machine hardware cost and deploys the application faster. EC2 Instance type comprises a different combination of CPU, memory, storage, and the networking capacity. It can be launched with a variety of operating system and configurations with security. It also provides firewall that enables to specify the protocols, ports, and source IP ranges that can reach your instances using security groups. The Amazon Simple Storage Service (Amazon S3) provides secure, durable, highly-scalable storage [16]. It is easy to store and retrieve data from anywhere with simple web service interface. With Amazon S3, you pay only for the storage that has been actually used.

- The implementation of the PrePost algorithm on cloud consists of following steps.

  Step 1: Initially, input dataset and Improve PrePost jar files located in Local System. Create an input bucket and upload these files to the AWS S3 bucket using simple AWS API with the provided access and secret key.
Step 2: Configure Hadoop cluster with different type of EC2 instances. The instance type is based on the requirement of computation. In Hadoop cluster, one instance is designated as Master node and others are designated as Slave shown in the architecture.

Step 3: After configuring Hadoop, Master EC2 instance fetch the dataset and jar files from S3 storage. AWS secret key and access key are added to hdfs-site file of Hadoop configuration for the communication between EC2 instance and S3 storage.

Step 4: Master EC2 instance node divides data sets into fixed size block and map to the different slave EC2 instance nodes in the Hadoop cluster.

Step 5: The Map function on slave EC2 instance nodes map input dataset to sets of key value pairs called intermediate results and applies the improved PrePost algorithms to produces the output on allocated dataset.

Step 6: The Master EC2 instance store the final result in HDFS file system in output bucket on S3 storage.

Step 7: The final result can be downloaded from s3 bucket to the local system with simple AWS API with the provided access and secret key.

The above architecture makes the system capable to handle increased computation and storage requirement. Hadoop cluster or single machine in Hadoop cluster can be scaled to manage computation need with AWS cloud. In some condition, Hadoop cluster crashes still data will be constitutively stored on S3 storage. This feature make the Hadoop cluster more fault tolerance and consistent. As Hadoop cluster run on AWS cloud, it can be integrated with AWS other services for better performance.

6. Experimental Results and Discussions

The experiment is performed on a Hadoop cluster in pseudo distributed mode. The system has a configuration with Ubuntu 14.4 and core i3 2.40GHz processor with 4GB RAM. Synthetic dataset is generated with the spawner data generator tool. T15I106D100K, T15I106D1000K and T15I106D2000K act as experimental data. T15I106D100K is a
small size dataset which contains 100000 transactions. T15I106D1000K and T15I106D2000K is a large dataset which contains 1000000 and 2000000 transactions respectively. In datasets, the average length of transaction is 15 and the number of unique items is 106. The datasets are space separated file with .txt extension. Before the improved PrePost algorithm, traditional PrePost data mining algorithms provide a good solution to find frequent itemsets. But when the massive data comes, the traditional PrePost algorithm goes out of memory and failed to find frequent items from massive datasets quickly.

In the experiment, comparing the run time of the Improved PrePost algorithm with PLFPG (a parallel linked list-based FPG) Algorithm [17]. Here, running time means the total execution time between input and output. Both the algorithms are based on Hadoop platform. In the above Fig. 3 (a), the vertical axis shows run time in milliseconds and the horizontal axis shows the different size of datasets. It can be seen that when the dataset is small, runtime of both the algorithm is near about the same. But, when dealing with large dataset, runtime of improved PrePost will become shorter as compared to PLFPG Algorithm. This shows that the improved PrePost gives better performance as compared to PLFPG Algorithm and can be used effectively for the analysis of large datasets. The Fig. 3 (b) shows the performance of the improved PrePost algorithm on AWS Cloud. In the experiment, the same three different size datasets, as mentioned above has been uploaded to S3 Storage. M4 type of EC2 instance has been used to configure Hadoop Cluster. The Hadoop cluster on EC2 instance fetches data From S3 storage, process it and stored the result to S3 Storage. The vertical axis shows run time in milliseconds and the horizontal axis shows the different size of datasets. Here, Comparing the run time of the Improved PrePost algorithm on local system with the improved PrePost algorithm on AWS Cloud. In the Fig. 3 (b), it is seen that the improved PrePost algorithm on AWS Cloud gives better performance as dataset size is increased. This shows that the improved PrePost with AWS gives a much better performance for large datasets. The above datasets has been used to compare the performance of improved PrePost algorithm with PLFPG algorithm. But, to evaluate performance of Improved PrePost algorithm with huge amount of
data, another synthetic large dataset T15I106D500K is generated with the spawner tool. T15I106D500K is a large size dataset which contains 5000000 transactions. It also contain the average length of transaction is 15 and the number of unique items is 106. In the above Fig. 4 (a), it shows the comparison of the Improved PrePost algorithm on local system with the Improved PrePost algorithm on AWS using T15I106D500K dataset. It is seen that Improved PrePost gives much better performance with large amount of data.

\[\text{Fig. 4. (a) Comparison of the Runtime of Improved PrePost with large amount of dataset; (b) Comparison of the Improved PrePost with different number of Map tasks}\]

In hadoop, the default number of map tasks depends on default number of HDFS blocks in input file. The size of T15I106D500K is 227540992 bytes and default blocks size is 67108864 bytes. So, the number of map tasks used is 3. The number of map tasks can be increased manually with conf.setNumMapTasks (int num) method. Fig. 4 (b) shows the runtime of the improved PrePost algorithm with different number of map tasks. It is seen that the runtime of Improved PrePost algorithm decreases as number of map tasks increases. But, instead of decreasing runtime of algorithm, it remains constant after particular point. In such situation, to decreases runtime, more nodes will be required to add into the Hadoop cluster instead of increasing number of map tasks per node.

With AWS, the Improved PrePost algorithm became more flexible as EC2 can be scaled up or scaled down to analyze different size of data. The performance of Hadoop also depends on the number of nodes in the cluster. So, when increases the number of nodes in the Hadoop clusters, the performance of the Improved PrePost algorithm will also improve.

7. Conclusion and Future Scope

This paper describes the implementation of the PrePost algorithm for data mining with Hadoop and architecture with public cloud. The Improved PrePost algorithm with Hadoop improves the shortcomings of the traditional PrePost algorithm. First, the paper describes step by step implementation of the PrePost algorithm on with Hadoop. Then, it describes the architecture of Hadoop clusters with AWS cloud and its different component. The Efficiency of the PrePost algorithm increases with general Tree method which reduces the time required for generating PPC tree.
Finally, the algorithm was validated with two different large and small datasets with a different environment. The result discussed in this paper shows that the PrePost algorithm with Hadoop gives 60% better performance as compared to PLFPG algorithm as dataset size increases. It also shows that the PrePost algorithm with AWS cloud make Hadoop cluster more flexible, scalable and gives better performance in the analysis of large datasets.

In future extensions of this work can be done by implementing the idea of the Improved PrePost algorithm for real time data processing. The framework like Apache Spark, Apache storm, Cloudera - Impala can be used to implement the algorithm with in-stream data processing. The efficiency and performance of the Improved PrePost algorithm can also be increased with integration of other AWS cloud services.

References