Apriori - A Big Data Analysis - A Review

Asif Ansari[#1], Ajit Parab[#2], Sachin Kadam [#3]

[#1]PG Scholar, Alamuri Ratnamala Institute of Engineering & Technology, Mumbai, India
[#2]H.O.D, Babasaheb Gawde Institute of Technology, Mumbai, India
[#3]Lecturer, Babasaheb Gawde Institute of Technology, Mumbai, India
[#1]ansariasif23@gmail.com

Abstract:- Apriori is designed to operate on databases containing transactions (for example, collections of items bought by customers, or details of a website frequentation). Other algorithms are designed for finding association rules in data having no transactions , or having no timestamps (DNA sequencing). Each transaction is seen as a set of items (an item set). Given a threshold C, the Apriori algorithm identifies the item sets which are subsets of at least C transactions in the database. Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time (a step known as candidate generation), and groups of candidates are tested against the data. The algorithm terminates when no further successful extensions are found.

Keywords: *Apriori, Apache Hadoop, MapReduce*

*****

INTRODUCTION

Data mining is the efficient discovery of previously unknown patterns in large datasets. It has attracted a lot of attention from both research and commercial communities for finding interesting information hidden in large datasets. One of the most important areas of data mining is association rule mining; its task is to find all subsets of items which frequently occur and the relationship between them by using two main steps: finding frequent item sets (core step) and generating association rules.

APRIORI- Map Reduce

Big Data has been generated in the areas of business application such as smart phone and social networking applications. Especially these days, the better computing power is more necessary in the area of Data Mining, which analyses tera or peta-bytes of data. Apriori-Map/Reduce Algorithm that implements and executes.

Apriori algorithm is a popular algorithm used to find the association rules. Apriori algorithm consists of repeating steps based on a criterion.

The algorithm uses the minimum support value to create a set of common repetitive elements as the criterion. The minimum support level depends on the context (Agrawal, 1994). If an item-set has a frequency larger than minimum support, then it is called as a frequent item-set. In addition, any subset if a frequent item-set is also a frequent item-set (Jin, 2003). See the following example (see Figure) (Zaiane, 2001). The database of transactions consist of the sets {1,3,4},{2,3,5},{1,2,3,5},{2,5}. Each number can be thought as a product or a term in a concrete experiment. The first step of Apriori is to count up the frequency (support) of each number (item) separately. The minimum support level as "1" for this example. Therefore, {4} is not frequent. So, we remove {4} from the first candidate item-set C1.

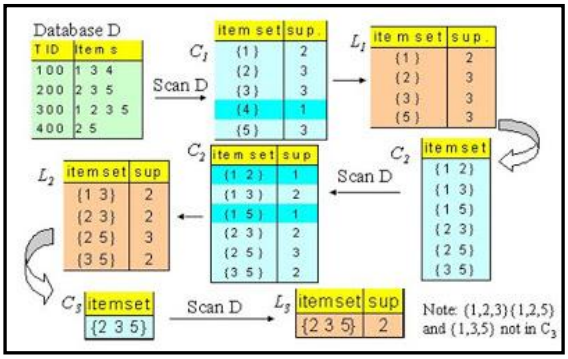

Figure 1

The next step is to generate C2 that is the item-set of all 2-pairs of the frequent items. In the same way, we remove {1, 2}, {1, 5} whose frequencies are "1". And then, we generate the tree of all possible sets. The third candidate item-set is {{2, 3, 5}}. Therefore, because of {2, 3, 5}'s support is 2, the last frequent item-set L3 is {{2, 3, 5}}. As a note, in the last scan, we did not include {1,2,3}, {1,2,5} and {1,3,5} in the C3 because in the previous scan we identified {1,2}, {1,5} items as infrequent, and {1,2}⊂{1,2,3}, {1,2}⊂{1,2,5}, {1,5}⊂{1,2,5}, and {1,5}⊂{1,3,5}. Now we can identify a association rule between the items of the last item-set.

Association rule generation is usually split up into two separate steps:

1. First, minimum support is applied to find all frequent itemsetsin a database.

2. Second, these frequent itemsets and the minimum confidence constraint are used to form rules.

While the second step is straight forward, the first step needs more attention.

Finding all frequent itemsets in a database is difficult since it involves searching all possible itemsets (item combinations). The set of possible itemsets is the power set over Iand has size 2n− 1 (excluding the empty set which is not a valid itemset). Although the size of the powerset grows exponentially in the number of items in I, efficient search is possible using the downward-closure propertyof support (also called anti-monotonicity) which guarantees that for a frequent itemset, allits subsets are also frequent and thus for an infrequent itemset, all its supersets must also be infrequent. Exploiting this property, efficient algorithms (e.g., Apriori and Éclat) can find all frequent itemsets.

$$
\begin{aligned}
&\text{Apriori}(T, \epsilon)\\
&L_1 \leftarrow \{\text{large } 1-\text{itemsets}\}\\
&k \leftarrow 2\\
&\textbf{while } L_{k-1} \neq \ emptyset\\
&C_k \leftarrow \{a \cup \{b\} | a \in L_{k-1} \wedge b \in \bigcup L_{k-1} \wedge b \notin a\}\\
&\textbf{for } \text{transactions } t \in T\\
&C_t \leftarrow \{c | c \in C_k \wedge c \subseteq t\}\\
&\textbf{for } \text{candidates } c \in C_t\\
&count[c] \leftarrow count[c] + 1\\
&L_k \leftarrow \{c | c \in C_k \wedge \ count[c] \geq \epsilon\}\\
&k \leftarrow k + 1\\
&\textbf{return } \bigcup_k L_k
\end{aligned}
$$

Apriori uses breadth-first search and a tree structure to count candidate item sets efficiently. It generates candidate item sets of length kfrom item sets of length k− 1. Then it prunes the candidates which have an infrequent sub pattern. According to the downward closure lemma, the candidate set contains all frequent k-length item sets. After that, it scans the transaction database to determine frequent item sets among the candidates.

Apriori, while historically significant, suffers from a number of inefficiencies or trade-offs, which have spawned other algorithms. Candidate generation generates large numbers of subsets (the algorithm attempts to load up the candidate set with as many as possible before each scan). Bottom-up subset exploration (essentially a breadth-first traversal of the subset lattice) finds any maximal subset S only after all 2 S− 1 of its proper subsets.

SAMPLE USAGE OF APRIORI ALGORITHM

A large supermarket tracks sales data by Stock-keeping unit (SKU) for each item, and thus is able toknow what items are typically purchased together. Apriori is a moderately efficient way to build a list of frequent purchased item pairs from this data. Let the database of transactions consist of the sets {1, 2, 3, 4}, {1,2,3,4,5}, {2,3,4}, {2,3,5}, {1,2,4}, {1,3,4}, {2,3,4,5}, {1,3,4,5}, {3,4,5}, {1,2,3,5}.

Each number corresponds to a product such as "butter" or "water". The first step of Apriori is to count up the frequencies, called the supports, of each member item separately:

ITEM	SUPPORT
1	6
2	7
3	9
4	8
5	6

Table 1

DOMAINS WHERE APRIORI IS USED

Application of the Apriori algorithm for adverse drug reaction detection. The objective is to use the Apriori association analysis algorithm for the detection of adverse drug reactions (ADR) in health care data. The Apriori algorithm is used to perform association analysis on the characteristics of patients, the drugs they are taking, their primary diagnosis, co-morbid conditions, and the ADRs or adverse events (AE) they experience. This analysis produces association rules that indicate what combinations of medications and patient characteristics lead to ADRs.

Application of Apriori Algorithm in Oracle Bone Inscription Explication

Oracle Bone Inscription (OBI) is one of the oldest writing in the world, but of all 6000 words found till now there are only about 1500 words that can be explicated explicitly. So explication for OBI is a keyand open problem in this field. Exploring the correlation between the OBI words by Association Rules algorithm can aid in the research of explication for OBI. Firstly the OBI data extracted from the OBI corpus are pre-processed; with these processed data as input for Apriori algorithm we get the frequent itemset. And combined by the interestingness measurement the strong association rules between OBI words are produced. Experimental results on the OBI corpus demonstrate that

 ISSN: 2321-8169
3517 - 3520

this proposed method is feasible andeffective in finding semantic correlation for OBI.

HADOOP

Hadoop is the parallel programming platform built on Hadoop Distributed File Systems (HDFS) for Map/Reduce computation that processes data as (key, value) pairs.

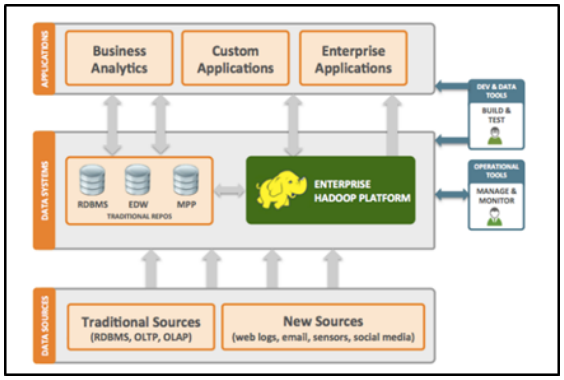

Figure 2: HADOOP archietecture

Hadoop has been receiving highlights for the enterprise computing because business world always has the big data such as log files for web transactions. Hadoop is useful to process such big data for business intelligence so that it has been used in data mining for past few years. The era of Hadoop means that the legacy algorithms for sequential computing need to be redesigned or converted d to Map/Reduce algorithms. Therefore, in this paper, a Market Basket Analysis algorithm in data mining with Map/Reduce is proposed with its experimental result in Elastic Compute Cloud (EC2) ans (Simple Storage Service) S3 of Amazon Web Service (AWS).

MAP/REDUCE IN HADOOP

Map/Reduce is an algorithm used in Artificial Intelligence as functional programming. It has been received the highlight since re-introduced by Google to solve the problems to analyze huge volumes of data set in distributed computing environment. It is composed of two functions to specify, “Map” and “Reduce”. They are both defined to process data structured in (key, value) pairs.

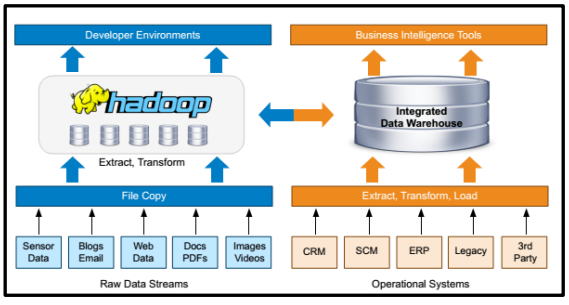

Figure 3: HADOOP and MapReduce

MAP/REDUCE IN PARALLEL COMPUTING

Map/Reduceprogramming platform is implemented in the Apache Hadoop project that develops open-source software for reliable, scalable, and distributed computing.

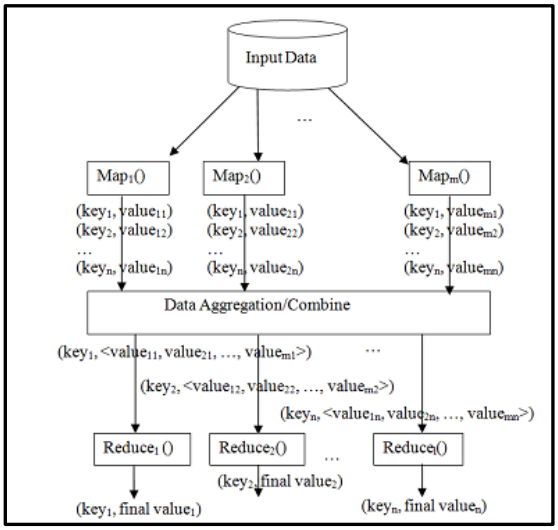

Figure 4: MapReduce Architecture

Hadoop cancompose hundreds of nodes that process and compute peta- or tera-bytes of data working together. Hadoop was inspired by Google's MapReduce and GFS as Google has had needs to process huge data set for information retrieval and analysis [1].

It is used by a global community of contributors such as Yahoo, Facebook, and Twitters. Hadoop’s subprojects include Hadoop Common, HDFS, MapReduce, Avro, Chukwa, HBase, Hive, Mahout, Pig, and ZooKeeper etc. [2].

THE ISSUES OF MAP/REDUCE

Although there are advantages of Map/Reduce, for some researchers and educators, it is:

1. Need tens-, hundreds -, or thousands-of-nodes to compose Hadoop Map/Reduce platform.
2. If using services of cloud computing, for example, AWS EC2, the overheads mainly come from I/O. That is, it takes long to upload big data to AWS EC2 platform or AWS S3, which is more than computing time.
3. A giant step backward in the programming paradigm for large -scale data intensive applications
4. Not new at all - it represents a specific implementation of well-known techniques developed tens of years ago, especially in Artificial Intelligence

5. Data should be converted to the format of (key, value) pair for Map/Reduce, which misses most of the features that are routinely included in current DBMS.
6. Incompatible with all of the tools or algorithms that have been built [4].

CONCLUSION

Hadoop with Map/Reduce motivates the needs to propose new algorithms for the existing applications that have had algorithms for sequential computation. Besides, it is (key, value) based restricted parallel computing so that the legacy parallel algorithms need to be redesigned with Map/Reduce.

Apriori algorithm can be used with FP growth tree in the future scope for the data mining.

The challenges include not just the obvious issues of scale, but also heterogeneity, lack of structure, error-handling, privacy, timeliness, provenance, and visualization, at all stages of the analysis pipeline from data acquisition to result interpretation. These technical challenges are common across a large variety of application domains, and therefore not cost-effective to address in the context of one domain alone.

References

[1] J. Dean and S. Ghemawa, "MapReduce: Simplified Data Processing on Large Clusters", Google Labs, OSDI 2004, (2004), pp. 137–150.

[2] Apache Hadoop Project, http://hadoop.apache.org/.

[3] W. Kim, "MapReduce Debates and Schema-Free", Coord, (2010) March 3

[4] Agrawal, R. and Srikant, R.. Fast algorithms for mining association rules in large databases. Proceedings of the 20th International Conference on Very Large Data Bases, VLDB, pages 487-499, Santiago, Chile, September 1994.

[5] Zaiane, O.R., University of Alberta. (2001). Web Technologies and Applications, Lecture Notes, Winter 2001.http://webdocs.cs.ualberta.ca/~zaiane/courses/cmput 499/slides/Lect10

[6] Jin, R., McCallen, S., Breitbart, Y., Fuhry, D. & Wang D., Department of Computer Science, Kent State University. Estimating the Number of Frequent Itemsets in a Large Database.

[7] Apache Hadoop project, http://hadoop. Apache .org/, March 2010.

[8] Patterson, J., "The Smart Grid: Hadoop at the Tennessee Valley Authority (TVA)," Cloudera Blog, http://www.cloudera.com/ blog/2009/06/smart-grid-hadoop-tennessee-valley -authority-tva/, June 2, 2009.

[9] White, T., Hadoop: The Definitive Guide. O'Reilly Media, May 2009.

[10] Lin, J. and Dyer, C., Data-Intensive Text Processing with MapReduce. To be published by Morgan and Claypool. Pre-press edition available at http://www.umiacs.umd.edu/~ jimmylin/MapReduce-book-20100219.pdf, February 2010.