IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 18, Issue 1, Ver. II (Jan – Feb. 2016), PP 06-10
www.iosrjournals.org

Mining Interesting Medical Knowledge from Big Data

Anand Kot, Nitish Kulkarni, Sushma Kaharat, Pranali Deole, Dr. K. Rajeswari

Abstract: *There are many real life applications where uncertain big data is generated. For example: medical data, stock exchange, social media and many more. Big data works on the data which is unstructured. Mining such a huge uncertain data is exhaustive due to the presence of values of items in every transaction in the uncertain data.*
Meaning there are many items present in big data with a high probability values with respect to its transaction. The value expresses the probability of that item to be present in a particular transaction in the big data. Mining such data is important to generate useful information.
The situations where, users not always interested in huge data, user may be interested in mining all frequent patterns from the uncertain big data, or, users may be interested only in a small portion of these mined patterns. Therefore data need to be processed, which produces results depending on user interest.
To reduce the computation and to focus the mining on user specified frequent pattern, we propose a solution that uses Map Reduce technique. Map reduce mines the uncertain big data for frequent patterns which satisfies user's need. These will result in eﬀectiveness of mining interesting patterns from uncertain big data and produce result depending on user interest.
This project aims to help medical practitioners to apply and use the obtained knowledge in their daily work to improve patient care and outcomes while managing costs.

I. Introduction

Now a day's large amount of data is generated from various hospitals and laboratories. The rate at which this data is generated is very high. This leads us to use make use of Big Data which refers tohighvelocity, high-veracity, high-value, and/or high-variety. It is beyond the capacity of the normal software to process such large amount of data within given time constraint. That is why new form of processing is require to handle such large amount of data. The processing should be such that it should help us gain insight of the data, and lead us to knowledge discovery. If the medical practitioners are able to get information from such large amount of data it will help them to take decisions for treatment of the disease. Also normal users can be benefitted by this program, because they can know about a disease event before it occurs. So it can help them to take various measures to avoid such disease. Which will eventually result in reduction of cost user for the treatment of the disease.

In recent years a programming model known as MapReduce [2] is been used by the researchers to process large amount of data. In MapReduce the Big data is processed in parallel by distributing its processing over commodity hardware [3], [4], [5]. MapReduce involvestwo key functions: map and reduce functions commonly used in functional programming languages such as LISP for list processing[1]:

- Map function listOfValues: which applies the function to each value in the listOfValues and returns the resulting list; and
- Reduce function listOfValues: In this function is applied to the result, to combine all the result and return the combined result.

The advantage of using MapReduce is that the user need not worry about the internal working. User only needs to specify map and reduce function. User can be relaxed about various operations such as partitioning the input data, scheduling and executing the program across multiple machines, handling machine failures, or managing inter-machine communication [1]. MapReduce is scalable , fault tolerant and it can process huge amount of data in parallel. It works on commodity hardware, so it is cheap.

One of the important task is association rule mining. In this interesting knowledge which is in the form of association such as whenever event A occurs event B occurs. This can help in medical field because it will be possible to find out relation between various diseases. Whenever one disease occur, how probable it is that some other disease also occur.The association rule mining [1] usually consists of two key steps, mining frequent patterns and formation of association rules (by using the mined frequent patterns as antecedents and consequences of the rules).

In many case only small portion of the mined information is useful to the users. But the algorithms mine all frequent patterns without considering the user need. This will result wastage of user time, because all the patterns which aremined are not of user's interest. Hence this project uses constraint pattern mining[6] which will mine only those patterns in which user is interested. This constraint is known as anti-monotonic constraint.

II. Background And Related Work

Anti- Monotonic constraint states that if a subset satisfies a specified constraint then its superset may or may not satisfy this constraint. But if subset does not satisfy this constraint then its superset will also not satisfy this constraint. But if a superset will satisfy the constraint then its subset will also satisfy the constraint [14]. This property can be used to reduce the search space.

The existing tree based algorithm such as UF-growth [9], CUF-growth [10] and PUF-growth [11] works in the following manner. They first scan the data, find the frequency count of each item, and then compare it with the support count. If the count is less than the support count then the item is pruned. The reason behind this is that its superset will also be infrequent. So there is no need to process such items. In the second scan frequent items are inserted into tree. At each step during the mining process, the frequent patterns are expanded recursively.

A pattern is said to be frequent pattern if its count is greater than user specified minimum support. The problem of frequent pattern mining from a database is to mine such patterns which satisfy user specified minimum support threshold.

Hadoop (Apache) is an open source software framework which is written in Java for processing and storage of distributed and system for large data sets on computer (clusters) built from commodity hardware.

Hadoop design in such a way that the failure in hardware i.e. individual machines or multiple machines are unexceptional and they should automatically handle framework software.

The core of Apache Hadoop consists of a storage part (Hadoop Distributed File System (HDFS)) and a processing part (MapReduce). Hadoop splits files into large blocks and distributes them amongst the nodes in the cluster. To process the data, Hadoop MapReduce transfers packaged code for nodes to process in parallel, based on the data each node needs to process. This approach takes advantage of data locality nodes manipulating the data that they have on hand—to allow the data to be processed faster and more efficiently than it would be in a more conventional supercomputer architecture that relies on a parallel file system where computation and data are connected via high-speed networking.

Hadoop has a special storage feature which is known as Hadoop distributed file system. Map and Reduce are used for processing the data.

III. Mathematical Equation

The mathematical model will try to answer some of the questions such as what is the relation between processing power and speed of the system. The terms used in the model are:

RT = Overhead + (one hour data processing time) *
Hours ….(1)
TT=Time(constant) required/spent in workflow +
Data Processing Speed per Hour * Total Data in terms of Hours ….(2)
Terms used:
RT->Runtime ….(3)
TT->Total time required ….(4)

Overhead is the constant amount of time that is required every time and which is not dependent on amount of data can be considered as overhead. The time taken to distribute data among the cluster can come in this category.

One hour data processing time means the time required to process one hour of data. But here overhead can be ignored. Here hour of data is the unit used to measure the amount of data.

Runtime refers to total amount of time required for processing complete data. Hours in the equation refers to amount of data which is given as input to the system.

Point at which the runtime of the workflow is equal to the number of hours of data.

TT=Total Data in terms of Hours ….(5)
TT=Time (constant) required/spent in ….(6)
workflow+Data Processing Speed per Hour*TT TT= Time (constant) required/spent in workflow/ (1

- Data Processing Speed per Hour) ….(7)

Why doesn't the speed of workflow double when the amount of processing power is doubled?

TT1 = Time (constant) required/spent in workflow
/ (1 - Data Processing Speed per Hour) ….(8)
TT2 = Time (constant) required/spent in workflow/
(1 - Data Processing Speed per Hour/2) ….(9)

Here TT1 is runtime before doubling the processing power and TT2 is runtime after doubling processing power. Now if we take ratio of these two terms we get a new term speedup.

TT2 / TT1 =2*(1 - Data Processing Speed per Hour) / (2 - Data Processing Speed per Hour) ...(10)

If we substitute Data processing speed per hour as 54 minutes which means that 54 minutes are required to process to process 1 hour of data. After calculating, we get result as 0.18. which means that new runtime is 0.18 times the old runtime. That means new speed is increase by 82%.

IV. Tools Used

A) Map-reduce[12]

1) Map-reduce is a model for processing ,storing and analyzing big data chunks.
2) Map-reduce has two phases: Mapping Phase and Reducing Phase.
3) They are basically use for large datasets processing
4) Mapper: It takes a data chunks from HDFS in (key, value) format and generates another (key1, value1) pairs for intermediate processing. The processed Values are passed to Reduce Function.
5) It can be represented as map::(key, value)=>(key1,value1
6) Intermediate values are merged together
7) Reducer: Intermediate values are provided with the help of iterator by which too large values fit in the memory.
8) It can be stated as follows reducer::(key1,list(value1))=>(key2,value2) 9)More than one output files may be written on HDFS

B) HDFS (Hadoop Distributed File System)

1) It stores the big data in block structure
2) HDFS is scalable and robust
3) HDFS is fault tolerant
4) It is best suited in fully distributed mode 5)Replication of Data helps the system to be fault tolerant

C) HADOOP[12]

1) Hadoop is nothing but a framework that permits processing of large datasets
2) Stand-alone environment has the limit to process big datasets so hadoop especially work on fully distributed mode
3) The master node is called as name-node
4) The slave nodes are called as data nodes
5) It is robust and can be scaled at any point of time
6) The Replication Property Of HDFS allows the system to be fault tolerant
7) Hadoop is open source as it is developed in java

The base Apache Hadoop framework is composed of the following modules:

1) Hadoop Common – contains libraries and utilities needed by other Hadoop modules;
2) Hadoop Distributed File System (HDFS) – a distributed file-system that stores data on commodity machines, providing very high aggregate bandwidth across the cluster;
3) Hadoop YARN – a resource-management platform responsible for managing computing resources in clusters and using them for scheduling of users' applications; and
4) Hadoop MapReduce – a programming model for large scale data processing.

V. Our Proposed Work

A) Data Flow Diagram

The diagram shows the flow of the data in the system. The administrator has the privilege to setthe number of clusters used for MapReduce operation and give training data set to the system. Then a model is constructed from the training data set. User gives his report as input to the system through the API. Then with the help of constructed model, the user query is answered. Here classifier and clustering is used because this project uses hybrid approach.

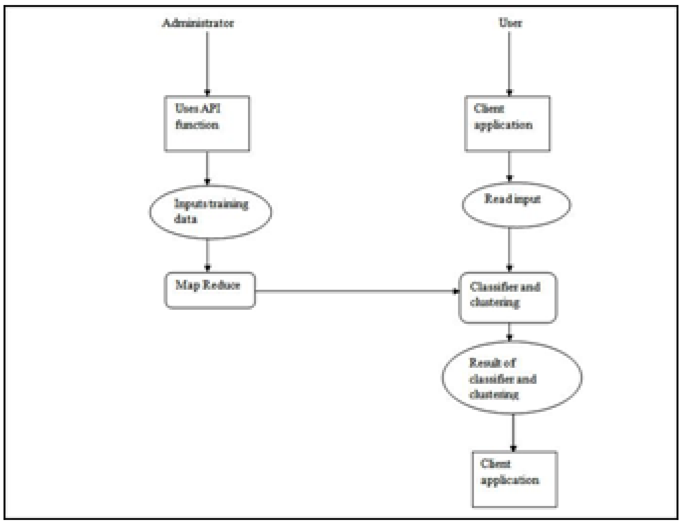

Fig. 1: Data flow diagram

The classifier and clustering can be elaborated as follows.
Algorithm K-means(K,D)

1) Choose points data points as initial Cluster mean.
2) Repeat

a. For every data point a from D
i. Calculate distance between a and each centroid.
ii. Put a in closest cluster

End for.
b. Compute mean for the current clusters.
3) Until there is no change in the cluster.

VI. Conclusion

Thus we propose a system to help medical practitioners to predict a disease and medical experts to take decision by providing results. Large amount of medical data is generated and it contain interesting knowledge. To address this interesting knowledge we propose a system which uses hadoop framework to process the data. In this project large amount of data is given as input to the system which is processed by HDFS. The model is constructed from training data sets. Whenever user inputs a query, it will be answered with the help of constructed model.

User might be interested only in small amount of patterns which are mined with respect to interest, so, to address this need of user, constrained mining is used to provide user specified data.

References

Basic format for books:
[1]. Carson Kai-Sang Leung Fan Jiang Department of Computer Science, University of Manitoba, Winnipeg, MB, Canada, “A Data Science Solution for Mining Interesting Patterns from Uncertain Big Data", 2014IEEE.
[2]. Kalavri, V. Brundza, and V. Vlassov, “Block sampling: efficient accurate online aggregation in MapReduce,” in Proc. IEEE CloudCom 2013, Volume 1, pp.250–257.
[3]. A. Cuzzocrea, C. K.-S. Leung, and R. K. MacKinnon, “Mining constrained frequent itemsets from distributed uncertain data,” *Future Generation Computer Systems*, 37, pp. 117–126, July 2014.
[4]. C. K.-S. Leung, R. K. MacKinnon, and F. Jiang, “Distributed uncertain data mining for frequent patterns satisfying antimonotonic constraints,” in *Proc. IEEE AINA Workshops 2014*, pp. 1–6.
[5]. M.J. Zaki, “Parallel and distributed association mining: a survey,” *IEEE Concurrency*, 7(4), pp. 14–25, Oct.–Dec. 1999.
[6]. R. T., Ng, L. V. S. Lakshmanan, J. Han, and A. Pang, “Exploratory mining and pruning optimizations of constrained associations rules,” in *Proc. ACM SIGMOD 1998*, pp. 13–24.

[7]. C. K.-S. Leung, “Frequent itemset mining with constraints,” in *Encyclopedia of database systems*, pp. 1179–1183, 2009.
[8]. Varun Chandola, Sreenivas R. Sukumar , Jack Schryver, Knowledge Discovery from Massive Healthcare Claims Data, KDD'13, August 11-14, 2013, Chicago, Illinois, USA., Copyright 2013 ACM
[9]. C.K.-S. Leung, M.A.F. Mateo, & D.A. Brajczuk, “A tree-based approach for frequent pattern mining from uncertain data,” in PAKDD 2008 (LNAI 5012), pp. 653–661.
[10]. C.K.-S. Leung & S.K. Tanbeer, “Fast tree-based mining of frequent itemsets from uncertain data,” in DASFAA 2012 (LNCS 7238), pp. 272– 287
[11]. C.K.-S. Leung & S.K. Tanbeer, “PUF-tree: A compact tree structure
[12]. Hadoop - the definitive guide, 2nd edition by Tom White for frequent pattern mining of uncertain data,” in PAKDD 2013 (LNCS 7818), pp. 13–25.

Basic format for books (when available online):

[1]. Author.(year,monthday).*Title.*(edition)[Typeofmedium].*v olume (issue).*Available: site/path/file *Example:*
[2]. J. Jones.(1991, May 10). *Networks.*(2nded.)[Online]. Available:http://www.atm.com