Survey of Big Data Analysis Using Predictive Analytics Algorithms and Its Use Cases

Nikita.V.Shahane
Department of Computer Engineering
Pimpri Chinchwad College of Engineering
Pune, India

Rutuja Pande
Department of Computer Engineering
Pimpri Chinchwad College of Engineering
Pune, India

S. R. Vispute
Department of Computer Engineering
Pimpri Chinchwad College of Engineering
Pune, India

***Abstract*— This paper presents Survey of Big Data Analysis using Predictive Analytics Algorithms and it's Use-Cases. Data continues a massive expansion in diversity, scale and complexity. Data underpin activities in all areas of society. Achieving the full transformative potential from the use of data in this increasingly digital world requires both new data analysis algorithms as well as new generation of systems and distributed computing environments to handle the dramatic growth in the volume of data. The scientific community can use Hadoop MapReduce on huge server farms that monitor natural phenomena and/or the results of experiments. Community needs to analyse data gathered by server farms monitoring search results and to identify potential terrorist threats. In this paper we propose big data platform that is built upon open source and built on Hadoop MapReduce also we are working for implementing parallel lingo algorithm for getting the distinct search on search engines.**

Keywords*— Predictive Analytics Algorithms, Big Data analytics, Data Mining, Hadoop, MapReduce, LINGO.

I. INTRODUCTION

Predictive Analytics is an area of data mining that deals with extracting information from data and using it to predict trends and behaviour patterns. Often the unknown event of interest is in future, but predictive analytics can be applied to any type of unknown whether it be in the past, present or future. Hadoop is a massive data storage and processing system. Hadoop has features like it is scalable distributed framework and fault-tolerant and Internet companies developed Hadoop to capture and analyse their data that they generate from their websites. It is able to store any kind of data in its native format. Hadoop can store megabytes of data, inexpensively. Robust and reliable data and handles hardware and system failures without losing data or interrupting data analyses. It has high performance and distributed data storage processing system. MapReduce and HDFS are the two important components in Hadoop.

II. PREDICTIVE ANALYTICS

Seven reasons, predictive analytics is needed.

A. Compete

Obtain the most powerful and unique competitive stronghold.

B. Grow

Growth in sales and retain customers competitively.

C. Enforce

Sustain business integrity by managing fraud.

D. Improve

Advances your core business capacity competitively.

E. Satisfy

Meet today's increasing consumer expectations.

F. Learn

Employ and implement today's most advanced analytics.

G. Act

It Render business intelligence and analytics truly actionable.

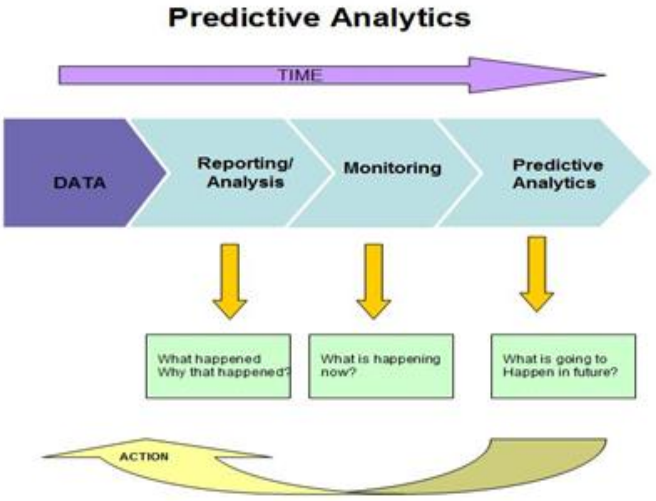


Fig 1.Predictive Analytics Modelling

III. ALGORITHMS USED

Algorithms perform data mining and statistical analysis in order to determine patterns and trends in data. The predictive analytics software solutions has built in algorithms such as regression, time series, outliers, decision trees, k-means, neural networks for doing this. Most of the software also provide involving many variables[5].

Different predictive analytics algorithms are as follows:

A. Time Series Algorithms

Time Series Algorithms which perform time based predictions. Example Algorithms are Single/Double/Triple Exponential Smoothing.

B. Regression Algorithms

Regression Algorithms which predicts continuous variables based on other variables in the dataset. Example algorithms are Linear/Exponential/Geometric/Logarithmic Regression, Multiple Linear Regression.

C. Associative Algorithms

Associative Algorithms which finds the frequent patterns in large transactional dataset to generate association rules. Example algorithms are Apriori, Partition, FP-Growth and ECLAT.

D. Clustering Algorithms

Clustering Algorithms which cluster observations into groups of similar groups. Example algorithms are k-means, DBSCAN, Fuzzy C-Means, Kohonen and TwoStep.

In this paper we are going to discuss about Parallel K-Means Algorithm using MapReduce.

IV. PARALLEL K-MEANS ALGORITHM BASED ON MAPREDUCE

In this section we represent the main design for Parallel K-Means(*PKMeans*) Algorithm based on MapReduce. At the first we give a brief overview about the *K-Means* Algorithm and analyse its parallel as well as serial parts of algorithms. Later, we explain about how the necessary computations can be formulized using map and reduce operation in detail[1].

1. *What is MapReduce?*

MapReduce is a programming model for processing and generating large data sets with a distributed, parallel algorithm on a cluster.

The model is inspired by the map and reduce functions commonly used in functional programming MapReduce, although their purpose in the MapReduce framework is not the same as in their original forms. The important contributions of the MapReduce framework are not the actual map and reduce functions, but the fault-tolerance and scalability achieved for a variety of applications by optimizing the execution engine once.

The frozen part of *MapReduce* framework is a large distributed cage. The hot spots, which the application defines[3][1]:

A. *Input Reader*
The *input reader* divides the input into appropriate size 'splits' (in practice typically 64 MB to 128 MB) and the framework assigns one split to each *Map*( ). The *input reader* reads data from stable storage and generates key/value pairs.

B. *Map Function*
The *Map* function takes a series of pairs of key and value, processes each, and generates zero or more output pairs. The input and output types of the map can be (and often are) different from each other.

C. *Partition Function*
Each *Map* function output is allocated to a particular *reducer* by the application's *partition* function for sharding purposes. The *partition( )* is given the key and the number of reducers and returns the index of the desired *reducer*.

D. *Comparison Function*
The input for each *Reduce* is taken from the machine where the *Map* ran and sorted using the application's *comparison* function.

E. *Reduce Function*
The framework calls the application's *Reduce* function once for each unique key in the sorted order. The *Reduce* iterates through the values that are associated with that key and produce zero or more outputs.

F. *Output Writer*
It writes the output of the *Reduce* to the stable storage.

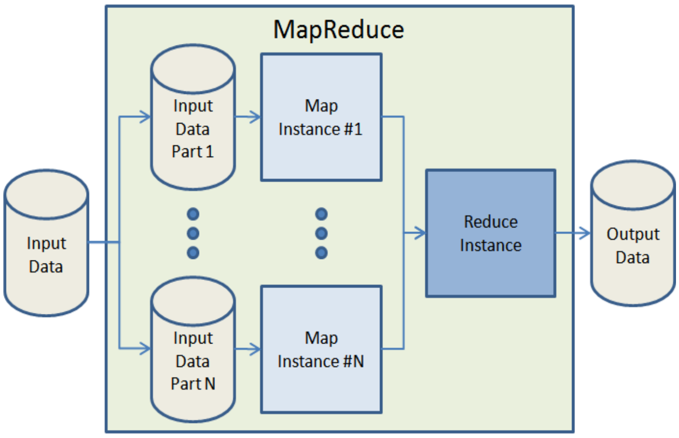


Figure 2. MapReduce

1. *Parallel K-Means Based on MapReduce*

Parallel k-means algorithm needs one kind of MapReduce job. Map function performs the procedure of assigning each sample to the closest centre while the reduce function performs the procedure of updating the new centres[1].

A. *Map Function*

The input dataset is stored on HDFS as a sequence file of <key, value> pairs, each of which represents a record in it. The **key** is the offset in bytes of this record to the start point of the data file, and the **value** is a string of the content of this record. The dataset record is split and globally broadcast to all mappers. Therefore, the distance computations are parallely executed. For each map task, PKMeans construct a global variant centres which is an array containing the information about centres of the clusters. It is given that a mapper can compute the closest centre point for each sample. The intermediate values are then composed of two parts: the index of the closest centre point and the sample information. The pseudocode of map ( ) is shown in Algorithm 1

.

i) <u>*Algorithm 1:*</u>

```
Input: Global variable centers, the sample value, the offset key.
Output: <key', value'> pair, where the key' is the index of the closest center point and value' is a string comprise of sample information

1.  Construct the sample instance from value;
2.  minDis = Double.MAX VALUE;
3.  index = -1;
4.  For i=0 to centers.length do
        dis=ComputeDist(instance, centres[i]);
        If dis < minDis {
                minDis = dis;
                index = i;
        }
5. End For
6. Take index as key';
7. Construct value' as a string comprise of the values of different dimensions;
8. output < key, value> pair;
9. End
```

Note: Step 2 and Step 3 initialize the auxiliary variable *minDis* and *index;* Step 4 computes the closest centre point from the sample, in which the function *ComputeDist* (*instance*, *centres*[*i*]) returns the distance between *instance* and the centre point *centres*[*i*]; Step 8 outputs the intermediate data which is used in the subsequent procedures.

B. *Combine Function*

After each map task, we apply a combiner to combine the intermediate data of the similar map task. The procedure cannot consume the communication cost. as the intermediate data is stored in local disk of the host. In the combine function, we partially sum the values of the points assigned to the similar cluster. In order to calculate the mean value of the objects for each cluster, we should record the number of samples in the similar cluster in the similar map task. The pseudocode for combine function is shown in Algorithm 2.

ii) <u>*Algorithm 2:*</u>

```
Input: key- index of the cluster, V- list of the samples assigned to the same cluster
Output: < key, value > pair, where the value' is a string comprised of sum of the samples in the same cluster and the sample number, the key' is the index of the cluster.

1. Initialize one array to record the sum of value of each dimensions of the samples contained in the similar cluster, i.e. the samples in the list V;
2. Initialize a counter num as 0 to record the sum of sample number in the similar cluster;
3. while(V.hasNext())
    {
        Construct the sample inst from V.next();
        Add the values of different dimensions of inst to the array
        num++;
    }
5. Take key as key';
6. Construct value' as a string comprised of the sum values of different dimension and num;
7. output < key, value> pair;
8. End
```

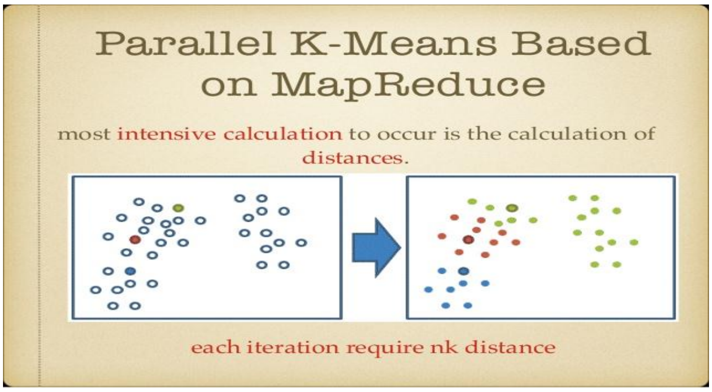
C. *Reduce Function*

The input of the reduce( ) is the data obtained from the combine( ) of each host. As described in the combine( ), the data includes partial sum of the samples in the similar cluster and the sample number. In reduce( ), we can sum all the samples and compute the total number of samples assigned to the similar cluster. Therefore, we can get the new centers which are used for next iteration. The pseudocode for reduce( ) is shown in Algorithm 3.

iii) <u>*Algorithm 3:*</u>

```
Input: key- index of the cluster, V- list of the partial sums from different host
Output: < key, value> pair, where the value' is a string representing the new centre, the key' is the index of the cluster.
```

```
1. Initialize one array record the sum of value of each dimensions of the samples contained in the similar cluster, e.g. the samples in the list V;
2. Initialize a counter NUM as 0 to record the sum of sample number in the similar cluster;
3. while(V.hasNext())
    {
        Construct the sample inst from V.next();
        Add the values of different dimension of inst to the array
        NUM += num;
    }
5. Divide the entries of the array by NUM to get the new centre's coordinates;
6. Take key as key';
7. Construct value' as a string comprise of the centre's coordinates;
8. output < key, value> pair;
9. End
```



Parallel K-Means Based on MapReduce

most intensive calculation to occur is the calculation of distances.

each iteration require nk distance

Figure 4. PK-Means Based on MapReduce

V. Big Data Hadoop Use Cases

We have been collecting data for years. In use-cases we learn how to use it to grow business and gain a competitive edge.

Some of the Big Data Use-Cases are mentioned below:

A. *Risk Modelling*

Hadoop is used to analyse the risk into their repositories. In a Hadoop cluster it contains more than a petabyte of data collected from multiple enterprise data warehouses.

Applying different techniques like, text processing, Pattern matching techniques, sentiment analysis, and graph creation. When there were some discrepancies in the identifying information stored automatic pattern matching is used to combine, digest and analyse the data.

The result of this analysis is a very clear picture of a risk modelling. The collection and combination of structured and complex data from multiple servers, and a tool of analytics that combine the data and look for patterns-apply broadly[6].

B. *Customer Churn Analysis*

The telecommunication company used Hadoop to combine traditional transactional and event data with social data coming from different websites. Creating a graph of the social network by examining the call logs and analysing it.

If the customer's social network were leaving, then likely your product does not satisfy the feature the customer wants. Analysing and combining the coverage maps with customer account data the cluster receives, the gaps in coverage affected churn of the analysis. Often any customer can use their handsets and frequently they may replace them with a new product enters in the market which gives more convenient and with extra additional features attracts the customers more than other devices. Hadoop allows the manufacturer to predict whether a particular customer was likely to change plans or providers. In this way the manufacturer gave the provider a much better measure of the risk that the customer would leave network investments to improve customer satisfaction and improved planning for new products[6].

C. *Ad-targeting*

Advertisement targeting is a special kind of recommendation engine. This engine helps to select best ads that attract the visitor to buy the particular product. Every advertiser is willing to pay a certain amount to have its ads on the Internet. Ad targeting creates a complex optimisation challenge and it must be understand by the systems user preferences and behaviour, estimating how a logged in user will be interested in different ads for displayed on the screen, and to choose the one that maximises revenue to both the advertising network and the advertiser. Managing the data by these systems is structured and simple . The exchanges, however, provide the services to a large number of advertisers and deliver the advertisements on a wide variety of Websites. This property must scale to all of end users browsing the web, loading pages and downloading the pages that must include advertising. This data volume is enormous and quick.

In this the optimization requires examining both the relevance of a given advertisement to a user and the collection of bids by different advertisers. This model uses large amount of visitor. This analytics required to make the correct choice for running them historical data on user behaviour to on the large dataset requires a large-scale, parallel system. Hadoop delivers much better-targeted advertisements by steadily refining those models and delivering better ads[6].

D. *Point of sale transaction analysis.*

A large retailer doing Point-of-Sale transactional analysis needed to combine larger quantities of PoS transaction analysis data with new and interesting data sources to forecast demand and improve the return that it got on its promotional campaigns. There retailer built a Hadoop cluster to understand its customers better and increased its revenues[6].

E. *Analysing network data to predict failure.*

In this use-case all large public powered companies combines their collected data. The data is then given to smart grid for processing with a map. The network data are predicted which have generates in form of grid. If the data is likely to fail then how the failure would affect their smart grid network. The detail study of this use case is done to avoid network failures[6].

F. *Threat Analysis*

Computers and on-line systems create new opportunities for criminals to act efficiently, anonymously and swiftly. Online businesses use Hadoop to monitor and combat criminal behaviour. Businesses have struggled with abuse, fraud and theft. On-line systems create new opportunities for criminals to attract the system in different ways they want. The online businesses uses Hadoop to monitor and combat criminal behaviour and to detect the theft before it harm the system [6].

G. *Trade Surveillance*

A trading merchant combines its data collected from different parties that participate in a trade. This data is a complex data that describes how those parties interact with one another. This combination allows the analyst to recognise unusual trading activity and to flag it for human review. Hadoop allows them to spot and prevent suspect trading activity .this issue is been concern for security purpose as no data should be accessed by non-user of that system [6].

H. *Search Quality*

Good search tools have been a boon to web users. The data available online is growing and organising the data has become increasingly difficult. Today the users are more likely to search for information with keywords. The user browse through folders looking for what he really needs. Efficient search tools are hard to build. the tool must store massive amounts of information, and much of it is complex text and multimedia files and to process those files to on other attributes for searches and the extract keywords. Scalable and flexible platform for indexing is demanding extract keywords .A search engine must be able to assess the intent and interests of the user when a search query arrives. Delivering meaningful results requires that the system make a good guess between the two[6].

I. *Data Sandbox*

Enterprises and data and warehouses often need a flexible and cost-effective way to explore, analyse and store new types of complex data. Many companies have created “data sandboxes” .in Hadoop the user can play with data, decide what to do with it and determine whether it should be added to the data warehouse. Analysts can look for new relationships in data and then mining it and using the techniques like natural language, machine learning and processing on it to get new insights [6].

VI. LINGO ALGORITHM AND OVERVIEW

The general idea behind LINGO is to first find meaningful descriptions of clusters, and then, based on the descriptions, determine their content. The algorithm must ensure that both the labels differ significantly from each other and at the same time cover most. It is possible to find such labels using the Vector Space Model along with the Latent Semantic Indexing(LSI) technique. To assign documents to the already labelled groups LINGO could use the Latent Semantic Indexing in the setting for which it was originally designed viz., given a query – retrieve the best matching do cuments. When a cluster label is fed into the LSI as a query, result contents of the cluster will be returned. This approach should take advantage of the LSI's ability to capture high-order semantic dependencies in the input collection. In this way not only would documents that contain the cluster label be retrieved, but also the documents in which the same concept is expressed without using the exact phrase [2].

```
/** Phase 1: Pre-processing */
for each document
{
        do text filtering;
        identify the document's language;
        apply stemming;
        mark stop words;
}
/** Phase 2: Feature extraction */
        discover frequent terms and phrases;
/** Phase 3: Cluster label induction */
        use LSI to discover abstract concepts;
        for each abstract concept
        {
        find best-matching phrase;
        }
        prune similar cluster labels;
Phase 4: Cluster content discovery */
for each cluster label
{
        use VSM to determine the cluster contents;
}
/** Phase 5: Final cluster formation */
        calculate cluster scores;
    apply cluster merging;
}
```

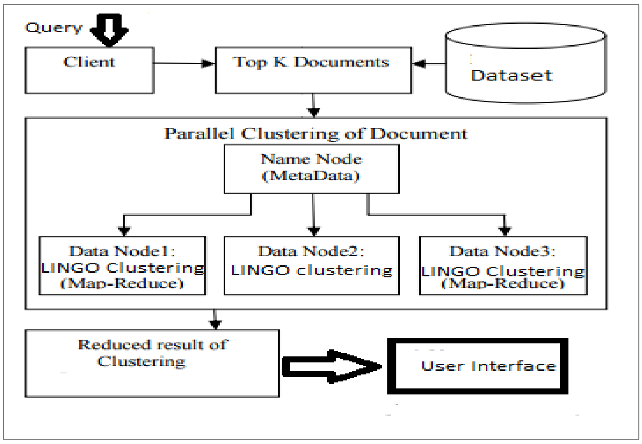

Query
Client
Top K Documents
Dataset
Parallel Clustering of Document
Name Node (MetaData)
Data Node1: LINGO Clustering (Map-Reduce)
Data Node2: LINGO clustering
Data Node3: LINGO Clustering (Map-Reduce)
Reduced result of Clustering
User Interface

Figure 5.Proposed work of Parallel Lingo Algorithm

VII. Search Categorisation Approaches

There are many techniques and algorithms exist for making searching efficient. Approaches such as Decision trees, Support Vector Machine, Neural Network, AdaBoost and Naïve Bayes . Several clustering all these techniques are available for categorisation query namely K-means, Suffix Tree Clustering (STC), Semantic Online Hierarchical Clustering (SHOC), Label Induction Grouping Algorithm (LINGO) etc. Here we are focussing on parallel Lingo clustering algorithm for making search less hectic. Clustering for Text Documents and multimedia files in parallel on Hadoop thus it makes a new approach ot searching the content on web and retrieving what the user wants. Clustering technique forms the groups of similar items from the input documents' set. The feature of high-quality clusters is that items into the same cluster are similar to each other, and these are unrelated for two dissimilar clusters. Let us discuss some approaches to the text clustering.

Hierarchic Agglomerative Clustering (HAC): Every step of the HAC algorithm merges an item/items and a cluster or two clusters' that are similar to each other into a new cluster and the association between items are symbolised in a genogram which is similar to tree.

"Suffix Tree Clustering(STC)","Semantic Hierarchical Online Clustering(SHOC)" and LINGO are created for the general query. The drawbacks of STC are overcome by SHOC.

LINGO algorithm is having lot of advantages over the other clustering algorithms such as it supports dynamic clustering as per the user query instead of static one, it identify cluster label first then assigns the document to that cluster, it is based on vector space model, it can be work for multiple languages and supports multiple keyword based searching.

VIII. Conclusion

Hadoop is widely used framework for storing big data. We take the benefits of the parallelism of MapReduce to design a parallel K-Means clustering algorithm based on MapReduce. This algorithm can automatically cluster the massive data, making full use of the Hadoop cluster performance. The survey of Hadoop use cases give the detail about using Hadoop
in different aspects and playing a vital role in everyday internet activities. The search engine use case made me to work for effective search of the fired query .The work is mainly focused on parallel clustering of lingo algorithm.

References

[1] Weizhong Zhao, Huifang Ma, and Qing He, "Parallel *K*-Means Clustering Based on MapReduce"

[2] Stainislaw Osinskl, "An Algorithm for clustering of web search results".

[3] Dweepna Garg, Khushboo Trivedi, B.B.Panchal, "A Comparative study of Clustering Algorithms MapReduce in Hadoop", Vol. 2 Issue 10, October – 2013, IJERT.

[4] "Fuzzy Ranking pf Financial Statements for Fraud Detection "Wei Chai, Bethany K. Hoogs and Benjamin T.Verschueren General Electric Global Research 1 Research Circle Niskayuna, NY 12309.

[5] DR. A. N. Nandakumar, Nandita Yambem,"A Survey on Data Mining Algorithms on Apache Hadoop Platform", Volume 4, Issue 1, January 2014, IJETAE.

[6] "Ten Common Hadoopable Problems Real-World Hadoop Use Cases" White Paper.

[7] "Authentication in an Internet Banking Environment towards Developing a Strategy for Fraud" Detection Kane Baxter Bignell Berwick School of Information Technology Monash University, Australia, bignell@netspace.net.au.