High Performance clustering for Big Data Mining using Hadoop

Rama Satish K V  
Assistant Professor  
RNSIT, Bangalore, India  

Dr. N P Kavya  
Professor & HR Manager  
RNSIT, Bangalore, India

Abstract
Now a day, organizations across public and private sectors have made a premeditated decision to big data into competitive advantage. The motivation and challenge of extracting value from big data is similar in many ways to the age-old problem of distilling business intelligence from transactional data. Hadoop is a speedily budding ecosystem of components based on big data Map Reduce algorithm and files system work for implementing Map Reduce algorithms in a scalable fashion and distributed on commodity hardware with clustering process. In this paper, we have focused on the distinct big data storage along with the web mining and how big data is a solution to many problems of the organizations. Big data not only focus to store and handle the large volume of data but also to analyzed and extract the correct information from the data in lesser time span. In this work, we will take Hadoop an open source framework for processing for massive datasets on cluster of different computers, which is shown with using the log file for extraction of information based on user query. This technique of Map Reduce and clustering technique will work together. Finally, we will use cluster technique for discussion on a use-case showing how enterprises can gain a competitive benefit by being early adopters of big data analytics.

Key words: Hadoop, Data mining, clustering, principal component analysis

I. INTRODUCTION

Cloud computing is emerging as the latest distributed computing paradigm and attracts increasing interests of researchers in the area of Distributed and Parallel Computing [1], Service Oriented Computing [2]. Though there is yet no consensus on what is Cloud, but some of its distinctive aspects as proposed by Ian Foster in [3] can be borrowed for an insight: “Cloud computing is a large-scale distributed computing paradigm that is driven by economies of scale, in which a pool of abstracted, virtualized, dynamically scalable, managed computing power, storage, platforms, and services are delivered on demand to external customers over the Internet [4].

Millions of user share cloud resources by submitting their computing task to the cloud system. Scheduling these millions of task is a challenge to cloud computing environment. Cloud computing has emerged as the advanced form of distributed computing, parallel processing and grid computing. As Clouds are designed to provide services to external users, providers need to be compensated for sharing their resources and capabilities [7] [14]. Since these computing resources are finite, there is a need for efficient resource allocation algorithms for the cloud platforms. Efficient resource/data allocation would help reduce the number of virtual machines used and in turn reduce the carbon footprint leading to a lot of energy saving [8] [15].

Scheduling in Map Reduce can be seen analogous to this problem [10]. If the scheduling algorithms are designed in a more intelligent way to avoid overloading any node and utilize most of the resources on a particular node, the runtime of the jobs could be lowered to a greater extent leading to a lot of energy saving [11]. These strategies considers different factors like cost matrix generated by using credit of tasks to be assigned to a particular resource, quality of Service (QoS) based meta-scheduler and Backfill strategy based light weight virtual machine scheduler for dispatching jobs, QoS requirements, heterogeneity of the cloud environment and workloads [12] [13]. Optimal resource allocation or task scheduling in the cloud should decide optimal number of systems required in the cloud so that the total cost is minimized and the SLA is upheld. Cloud computing is highly dynamic, and hence, resource allocation problems have to be continuously addressed, as servers become available/non-available while at the same time the customer demand fluctuates. Thus this study focuses on scheduling algorithms in cloud environment considering above mentioned characteristics, challenges and strategies [18].

Map Reduce is a programming model and an associated implementation for processing and generating large data sets [19]. It enables users to specify a map function that processes a key/value pair to generate a set of intermediate key/value pairs, and a reduce function that
merges all the intermediate values associated with the same intermediate key [17]. Map Reduce is used in Cloud Computing in the beginning [16]. It is initiated by Google, together with GFS and Big Table comprising backbone of Google’s Cloud Computing platform. Map Reduce has achieved increasing success in various applications ranging from horizontal and vertical search engines to GPU to multiprocessors, e.g... Recently, Map Reduce has become a standard programming model for large scale data analysis. It has seen a tremendous growth in recent years especially for text indexing, log processing, web crawling, data mining, machine learning etc.

II. RELATED WORKS

Here, a review of some of the work presented for task scheduling in cloud computing. Rosmy C Jose et al. [21] presented an approach to scheduling scientific workflows on clouds. Scheduling multitask workflows in virtual clusters is a NP-hard problem. Excessive simulation time in weeks of time may be needed to produce the optimal schedule using Monte Carlo simulations. To reduce this scheduling overhead is necessary in real-time cloud computing. They presented a new workflow scheduling method based on iterative ordinal optimization (IOO). Victoria López et al. [22] presented a heuristic task scheduling algorithm called Balance-Reduce (BAR), in which an initial task allocation was produced at first, then the job completion time can be reduced gradually by tuning the initial task allocation. By taking a global view, BAR can adjust data locality dynamically according to network state and cluster workload. The simulation results show that BAR was able to deal with large problem instances in a few seconds and outperforms previous related algorithms in term of the job completion time.

Sara del Río et al. [23] presented, an idealized Hadoop model was presented to investigate the Hadoop task assignment problem. It was shown that there was no feasible algorithm to find the optimal Hadoop task assignment unless P = NP. Assignments that are computed by the round robin algorithm inspired by the current Hadoop scheduler are shown to deviate from optimum by a multiplicative factor in the worst case. A flow-based algorithm was presented that computes assignments that are optimal to within an additive constant. Qingchen Zhang, et al. [24] presented the scheduling problem in hybrid clouds presenting the main characteristics to be considered when scheduling workflows, as well as a brief survey of some of the scheduling algorithms used in their systems. To assess the influence of communication channels on job allocation, they compared and evaluated the impact of the available bandwidth on the performance of some of the scheduling algorithms.

Qingchen Zhang et al. [24] presented a Revised Discrete Particle Swarm Optimization (RDPSO) to schedule applications among cloud services that takes both data transmission cost and computation cost into account. Experiment is conducted with a set of workflow applications by varying their data communication costs and computation costs according to a cloud price model. Comparison is made on make span and cost optimization ratio and the cost savings with RDPSO, the standard PSO and BRS (Best Resource Selection) algorithm. Experimental results show that the proposed RDPSO algorithm can achieve much more cost savings and better performance on make span and cost optimization. Bogdan Ghit et al. [25] presented an integer linear program (ILP) formulation for the problem of scheduling SaaS customer’s workflows into multiple IaaS providers where SLA exists at two levels. In addition, they presented heuristics to solve the relaxed version of the presented ILP. Simulation results show that the proposed heuristics are effective when deadlines are larger.

III. RESEARCH METHODOLOGY

High dimensional data introduce several problems to traditional statistical analysis. As previously mentioned, computation time increases more rapidly with p than with n. For combinatorial and projection pursuit algorithms, this increase is of sufficient magnitude that it is not clear how such methods can be made feasible for high dimensional data.

a. Research Objective

- To maintain the large dataset for classification and clustering
- To maintain all data process with map-reduction and hadoop
- To efficiently collect and preprocess complex data sets that includes the huge quantities of data.
- To improve the quality and efficiency of care while cultivating patient centricity through engagement and product personalization.
- To adapt to the changing big data processing such as Hadoop, which uses the map-reduce paradigm.
b. Problem definition and Contribution of the paper

A classical supervised clustering problem consists of finding a function which, taking a set of random feature variables as arguments, predicts the value of a one-dimensional discrete random class variable. There exist scenarios, however, where more than one class variable may arise, so the extension of the classical problem to the multidimensional class variable case is increasingly earning the attention of the research community.

Data mining, also popularly known as Knowledge Discovery in Database, refers to extracting or “mining” knowledge from large amounts of data. Data mining techniques are used to operate on large volumes of data to discover hidden patterns and relationships helpful in decision-making. While data mining and knowledge discovery in database are frequently treated as synonyms, data mining is actually part of the knowledge discovery process. The sequences of steps identified in extracting knowledge from data are shown in Figure 1.

Recently, various researchers present several algorithms for Facebook three product Facebook three product classification based on classification methods. But, the challenge is not only in finding the Facebook three product words and also, how the dimensionality and scalability is taken into consideration for Facebook three product classification because, in reality, the processing is with a large and high dimensional data. So, (i) curse of dimensionality By handling these criteria, an Facebook three product Facebook three product classification technique is urgently needed for improving the classification accuracy. By solving the above challenge in this research, the feature selection is the main aspect for our research. The feature selection method can solve the curse of dimensionality by identifying the suitable features.

- Curse of dimensionality problem is solved by identifying the suitable features. Here, firefly algorithm is used for this suitable feature selection process.

c. Efficient Facebook three product Facebook three product classification through proposed feature extraction algorithm for naïve classifier

The ultimate target of this research is to design and develop a technique for Facebook three product classification using SVM classifier. The SVM algorithm Facebook three product filtering is a probabilistic classification technique of Facebook three product filtering which is based on SVM theorem with naïve independence assumptions. Let us consider each of the Facebook three product can be illustrated by a set of features (attributes) \( \{a_n\} \), where \( 1 < n > N \). Filtering of Facebook three product with SVM by considering of all features is very difficult also it need more time. In order solve this problem in this paper; we propose an efficient algorithm to select the significant features from the available to filter the Facebook three product in efficient manner. The overall model of the proposed-mail Facebook three product classification system is given in the following figure 1 and each part of the framework is elucidated concisely in the following sections.

![Figure 1: System Architecture](image)

d. Facebook Dataset

The words dataset is taken from the UCI machine learning repository and which is formed by Mark Hopkins, Erik Reeb, George Forman, Jaap Suermondt. Hewlett-Packard Labs. Their collection of Facebook three product words came from their postmaster and individuals who had filed Facebook three product and their collection of non-Facebook three product words came from filed work and personal words, and hence the word ‘George’ and the area code ‘650’ are indicators of non-Facebook three product. This Facebook three product dataset consists of 4601 instances and 58 attributes in which 57 continuous real attributes and 1 nominal attribute. From the Facebook three product dataset, 80% (3681 instances) of instances are taken for finding the significant features among the available (58) attributes for the training process and the other 20% (920 instances) of instances are taken for the testing process with the significant attributes. The descriptions of those attributes are given in the following table 1.
Table 1: Description about the attributes of Facebook three product dataset

<table>
<thead>
<tr>
<th>Attribute number</th>
<th>Category of attribute</th>
<th>Description of attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1 to A10</td>
<td>Word_freq_WORD</td>
<td>percentage of words in the Like that match WORD</td>
</tr>
<tr>
<td>A11 to A20</td>
<td>char_freq_CHAR</td>
<td>Percentage of characters in the Comment that match CHAR</td>
</tr>
<tr>
<td>A21 to A30</td>
<td>capital_run_length-_average</td>
<td>average length of uninterrupted sequences of Follower</td>
</tr>
</tbody>
</table>

IV. PRINCIPAL COMPONENT ANALYSIS (PCA)
The most common derivation of PCA is in terms of a standardized linear projection, which maximizes the variance in the projected space [29]. For a set of observed dimensional data vectors, \( t_n \), where \( n \in \{1, \ldots, N\} \) and \( q \) the principal axes \( w_j = 1, \ldots, q \), are those ortho-normal axes onto which the retained variance under projection is maximal. It can be shown that the vectors \( w_j \) are given by the \( q \) dominant eigenvectors those with the largest associated Eigen values of the sample covariance matrix \( S = E - [(t - \mu)(t - \mu)^T] \) such that \( Sw_j = \lambda_j w_j \).

The \( q \) principal components of the observed vector \( t_n \) are given by the vector \( x_n = W^T (t_n - \mu) \), where \( W^T = \{w_1, \ldots, w_q\}^T \). The variables \( x_j \) are then decorrelated such that the covariance matrix \( E[xx^T] \) is diagonal with elements \( \lambda_j \). The objective of PCA is to perform dimensionality reduction while preserving as much of the randomness in the high-dimensional space as possible.

Consider a \( M \times N \) matrix with random vector, represented as a linear combination of ortho-normal basis vectors \( C = \{a_1, a_2, \ldots, a_n\} \). The next step is to find the covariance matrix \( C_v \) from the empirical mean values which can be obtained by calculating through each dimension of the \( M \times N \) matrix, where \( m \) ranging from 1, \ldots, 5.

\[
e(m) = \frac{1}{N} \sum_{n=1}^{N} D[m, n]
\]  

(1)

Where, \( D \) is the matrix of size \( M \times N \). Then, the derivations of the means are calculated and stored in a matrix, \( \text{der}V \).

\[
\text{der}V = D - u.h, \ h = 1, \ldots, N
\]  

(2)

Thus, the covariance matrix \( C_v \) can be given by,

\[
C_v = \frac{1}{N} \sum \text{der}V . \text{der}V^T
\]  

(3)

The eigenvector vector of the covariance matrix is to be calculated and to be stored in a matrix \( V \). After finding the eigenvalues from the matrix \( C_v \), we calculated another diagonal matrix of \( C_v \) which, contains the eigenvalues.

\[
\text{diag}(C_v) = V^{-1} C_v V
\]  

(4)

From the above-obtained data, we construct another matrix, \( W \) which contains the eigen values, represented as,

\[
W(x, y) = V(x, y)
\]  

(5)

Where, \( W \) is a \( M \times L \) matrix, the field \( L \) is the columns of matrix \( V \). The next step of the dimensionality reduction process is to calculate the empirical standard deviation, \( \psi \) which contains the square root of each element along the main diagonal of the covariance matrix \( C_v \) and then, this value is supplied to calculate the Z-score matrix.

\[
\psi(m) = \sqrt{C_v[x, y], x = y = m}
\]  

Thus, we get the reduced dimension matrix as \( R \), which is the dot product of the conjugate of the matrix \( W \), which having dimension of \( M \times L \), and the Z-score matrix. Thus dimensionality reduction can be represented as

\[
R = W^T . Z
\]  

(6)

(7)
a. Dimensionality Reduction Using PCA

The clustering of high dimensional data deals with many interruptions. So it is quite hard to implement a clustering algorithm into a high dimensional data with its actual dimension. The most suitable way is to reduce the dimension of the data for the smooth processing of clustering. A number of methods are used for the dimensionality reduction and those methods are effective to some extends. In the proposed method, we also use a dimensionality reduction algorithm. The PCA algorithm is taken into concern for the proposed approach. This section gives complete idea of the PCA algorithm. Let us consider the high dimensional data with $N \times M$ dimension. According to the definition of the PCA algorithm, we calculate the empirical mean value, $e$, for the high dimensional data $D$. The covariance matrix of the data $D$ is calculated by deriving the $e$ values of the data.

$$D_{cov} = \frac{1}{N} D_e D_e^T$$

(8)

The above expression represents the covariance matrix, $D_{cov}$ and the derived value and the transpose of the derived values of empirical means values of the data $D$, which are represented by $D_e$ and $D_e^T$. The empirical standard mean is calculated by virtue of the covariance matrix and the diagonal matrix generated from the data. The empirical standard mean of the data $D$ can be given by the following equation.

$$\psi(D) = \sqrt{D_{cov}[x,y]}$$

(9)

Where, $x, y$ are the subset of row values of the $M \times N$ data matrix. The $\psi(D)$ represents the empirical standards mean value of the data $D$. The reduced dimension can be attained for the high dimensional data by calculating the $Z$-score of the data using the empirical standard mean $\psi(D)$. The values are calculated as per the definition of the PCA algorithm and the reduced matrix $R$ is obtained as the dot product of the $Z$-score matrix and the conjugate of the matrix $W$, which obtained from the covariance matrix.

$$R = W.Z$$

The reduced matrix is then considered for the further processing of our proposed approach. The reduced dimensionality helps a smooth processing of the proposed approach.

b. Finding attribute weightage values

The reduced matrix obtained from the previous step is used to find the weightage of the attributes that are the most relevant part, which we are taken for the analysis of the proposed clustering algorithm. The attribute weightage calculation is the separate measure, which we consider as the catalyst for our proposed approach. At first, the discretization has done for reduced matrix using the attribute threshold so that the matrix is converted to the discretized value. Then, the attribute weightage process is carried out. The attribute and data representation obtained from the previous step is represented in figure 3.

In order to increase the effectiveness of the algorithm, we find the weightage value for the attributes after the discretization. We have adopted a method in which, the weightage for each attribute $W(A_j)$ is calculated using the following expression.

$$W(A_j) = \sum_{i=1}^{k} \frac{freq(a_i) \times (freq(a_i) - 1)}{N_a^j (N_a^j - 1)}$$

(10)

Where, $A_j$ represents the attributes, $a_i$ denotes the unique values in the attribute $A_j$, $freq(a_i)$ signifies the frequency of the $a_i$, $N_a^j$ represents the total number of
values in the $A_j$ and $k$ is the number unique values in
the attribute $A_j$.

Consider the following example that is the reduced matrix after discretization,

<table>
<thead>
<tr>
<th>A1</th>
<th>A2</th>
<th>A3</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>C</td>
<td>D</td>
</tr>
<tr>
<td>B</td>
<td>C</td>
<td>C</td>
</tr>
<tr>
<td>A</td>
<td>C</td>
<td>E</td>
</tr>
</tbody>
</table>

Figure 4. Example data matrix for attribute weightage calculation

Consider the weightage value for attribute A1, the value can be generated by the following process.

\[ W(A_1) = \frac{2 \times 1}{3 \times 2} + \frac{1 \times 1}{3 \times 2}; \quad W(A_1) = \frac{1}{2} \] (11)

In the similar way, we calculate weightage of all other attribute that constitute the data matrix. The weightage value is then passed to the bisecting k-means algorithm for the improving the effectiveness of the proposed algorithm.

\[ W(A_2) = \frac{3 \times 2}{3 \times 2}; \quad W(A_2) = 1 \] (12)

\[ W(A_3) = \frac{1 \times 0}{3 \times 2} + \frac{1 \times 0}{3 \times 2} + \frac{1 \times 0}{3 \times 2}; \quad W(A_3) = 0 \] (13)

3.4. Clustering using SVM

SVMs pertain to the generalized linear classifier’s family. SVMs are also regarded as a special case of Tikhonov regularization. A peculiar property is that they lessen the empirical clustering error and increase the geometric margin at the same time. Therefore, they are termed as maximum margin classifiers. In this SVM, we utilize the vector $Pr$ for the training process in which identifies the abnormal word. This vector contains the parameters such as mean, standard deviation, maximum amplitude value and its id, minimum amplitude value and its id, MFCC length for both the normal and abnormal words are utilized and therefore 14 inputs as total. Here we detail the SVM clustering for the abnormality of words. The following equation is the SVMs’ objective function, which may identifies the support vector for the clustering

\[ Out = \sum_i w_i * K(s_i, pr) + b_i \] (14)

where

- $s_i$ = support vectors
- $w_i$ = weight
- $b$ = bias
- $pr$ = vectors for classification
- $k$ = kernel function

The aforementioned equation is the objective function utilizes an optimization method to identify the support vectors, weights and bias for classifying the vector $Pr$ where $k$ is a kernel function. In the case of a linear kernel, $k$ is the dot product.

\[
\begin{array}{c}
\text{If } out \geq \text{thresh} \text{ then} \\
\quad \text{Normal} \\
\text{Else} \\
\quad \text{Abnormal} \\
\text{End If}
\end{array}
\]

The above step indicates that the value of the variable $out$ than the threshold value then the class falls into the normal category else, it may falls into the abnormal category. SVM contains error also the error minimization function is as follows

\[ \arg\min中最小值 \sum_{x=0}^{n-1} v_x + 0.5 \lambda \lambda^T \] (15)

with the following constraints,

\[ cl \hat{k}(pr_x) + c) \geq 1 - v_x \] (16)

\[ v_x \geq 0 \] (17)

In Eq. (4), $pt$ is the penalty constant, $\nu$ is a parameter that handles the data and $\tilde{\lambda}$ is a matrix of coefficients. In the constraints given in Eq. (16) and (6), $cl_x$ is the class label of the $x^{th}$ dataset, $c$ is a constant and $k$ is the kernel that transforms the input data to the feature space. Hence, by minimizing the error function, the SVM learns the training dataset $Pr$ well and so that it can classify the vector that is similar to the training set. Once the errors minimized to a minimum value and hence we obtain the abnormal words separately.
Table 2: Multi SVM pseudo Code

```plaintext
Number of Class: N
Train Data
Test Data
Label

For n = 1: N
    Train = Train data (n);
    Group = find (label == n)
    Method (n) = svmtrain (Train, group);
End

For t = 1: N
    TestResut;t = svmclassify (method, test data (t))
End
```

V. EXPERIMENTAL SETUP

The trending topics application identifies recent trends on the web by periodically launching Cloudera’s Distribution for Hadoop to process Wikipedia log files. The daily pageview charts on the site were created by running an initial MapReduce job on GB’s of hourly traffic logs collected from Wikipedia’s mysql dumps. We run a job on the trendingtopics.org server to fetch the latest log files every hour and store a copy on mysql db for processing by Hadoop and we use Sqoop to load the data into hive.

For performing the big data experiments, setup of Hadoop data cluster and Hadoop Distributed File System (HDFS) for storage was used. Before moving to multi-node cluster, single node cluster was first configured and tested. We configured our cluster to run map reduce jobs for finding trends in log data with Hive. Input and output data for the Map/Reduce programs is stored in HDFS, while input and output data for the data-parallel stack-based implementation is stored directly on the local disks. The software used to setup these hosts are Sun Java 1.7, Cloud era quick start vm 5.1.0 and for visualization layer we have used Node.js.

VI. RESULTS

a. Data Sets and Descriptive Analysis

The selected high dimensional big data utilized various measures and different time intervals and session lengths. Hence, the obtained data were not immediately comparable. To solve this issue, the data were standardized. Using this method, we conducted a series of ordinary participant-specific regression analyses, whereby SVM was predicted by the condition. That way, the root mean squared errors were estimated. Subsequently, the raw data of each participant were divided by the participant’s root mean squared error in order to get standardized data. Furthermore, before conducting the meta-analysis, we carried out a descriptive analysis to get more insight into the data. The obtained frequencies, means, standard deviations, ranges, and correlations of possible moderators.

b. Selected Drinking Item Clustering Results

From the whole selected set of data, some seats are taken for training and part of it are taken for testing purpose and then this procedure is repeated for the whole Facebook database. The results are evaluated with both the training algorithms on all the combination of sets, but due to space constraints, only some of the results are listed and compared. The specifications of selected drinking item are given in table 1-3 with different product with different branches. From the results, SVM algorithm proves superior, as it is taking advantage of clustering methodology and searching technique for clustering.
c. Map Resource Utilization:
The map resource utilization of Hadoop and our proposed model is been plotted in the above graph. We have considered a maximum of 8 maps. Here we have taken the execution time by varying the map size and the analytical result proves that the proposed resource utilization time is reduced by 35 Sec from 57 Sec approximately over Hadoop.

VII. CONCLUSION

In this paper, we have presented an efficient technique to classify the data using SVM classifier. Proposed technique is comprised into two phase, (i) Map reduce framework for training with PCA and (ii) Map reduce framework for testing SVM. Initially, the input Facebook three product data is given to the feature selection to select the suitable feature for big data classification. The traditional some existing algorithm is taken and the optimized feature space is chosen with the best fitness. Once the best feature space is identified through SVM algorithm, the big data classification is done using the SVM classifier. Here, these two processes are effectively distributed based on the concept given in Map-Reduce framework. The results for the reduction output are validated through evaluation metrics namely, sensitivity, specificity, accuracy and computation time. For comparative analysis, proposed big data classification is compared with the existing works such as SVM and PCA for Facebook datasets.

REFERENCES

of the IEEE 23rd International Conference on Data Engineering, pp. 6-20, Istanbul, Turkey, 2007.


