The Big Data Classification using Infrequent Principle Component Analysis on Map Reduce Paradigm

[1]G.Yuvarani,[2]K.Saranya
[1]M.E II year,[2]Assistant professor
[1,2]Department of Computer science and Engineering,
Jayshriraam Group of Institutions,Tirupur

Abstract --Big Data is usually defined by three characteristics called 3Vs (Volume, Velocity and Variety). It refers to data that are too large, dynamic and complex. In this context, data are difficult to capture, store, manage and analyze using traditional data management tools. Thus, the new conditions imposed by Big Data present serious challenges at different level, including data clustering. This work makes a concise synthesis related to clustering to categorize efficiently using PCA based classification model on the map reduce framework. The Map Reduce framework is modelled using a novel technique named as Infrequent Principle component Analysis (IPCA) based Classification algorithm. The proposed technique is capable of classifying and indexing the large volume and high dimensional data. It works using data reduction methods like Singular value decomposition technique to eliminate the unwanted or less relevant attributes or fields in the data Matrix taken distance computation. The Infrequent PCA is capable of partitioning and extracting the feature for data labelling of classes. Usually class contains the set of relevant features. Load balancing is also managed by efficient configuring of the proposed algorithm with map reduce function. Experimental studies on various big datasets have been conducted. The performance of IPCA is judged in comparison with the KNN methods implemented on the map reduce framework. The comparative results are reported in terms of time and space complexity, run time and measure of clustering quality, showing that proposed method is able to run in less time without compromising the clustering quality.

Index Terms— Principle Component Analysis, Infrequent Principle Component Analysis, Clustering, KNN, Matrix, Load balancing, Indexing

I.INTRODUCTION

A.Distributed large Scale Processing

Large scale data analysis is the process of applying data analysis techniques to a large amount of data, typically in big data repositories. It uses specialized algorithms, systems and processes to review analyze and present information in a form that is more meaningful for organizations or end users. It encompasses a series of different tools and systems to process big data. Typically, large scale data analysis is performed through two popular techniques: parallel database management systems (DBMS) or MapReduce powered systems. The parallel DBMS system requires the data to be in a schema that supports DBMS,whereas the MapReduce option supports data in any form. Moreover, the data extracted or analyzed in large-scale data analysis can be displayed in various different forms, such as tables, graphs, figures and statistical analysis, depending on the analysis

system. Data analytics refers to qualitative and quantitative techniques and processes used to enhance productivity and business gain. Data is extracted and categorized to identify and analyze behavioural data and patterns, and techniques vary according to organizational requirements. Data analytics is primarily conducted in business-to-consumer (B2C) applications. Global organizations collect and analyze data associated with customers, business processes, market economics or practical experience. Data is categorized, stored and analyzed to study purchasing trends and patterns.

B. Big Data Analytics

Big data analytics refers to the strategy of analyzing large volumes of data, or big data. This big data is gathered from a wide variety of sources, including social networks, videos, digital images, sensors, and sales transaction records. The aim in analyzing all this data is to uncover patterns and connections that might otherwise be invisible, and that might provide valuable insights about the users who created it. Through this insight, businesses may be able to gain an edge over their rivals and make superior business decisions. Big data analytics allows data scientists and various other users to evaluate large volumes of transaction data and other data sources that traditional business systems would be unable to tackle. Traditional systems may fall short because they're unable to analyze as many data sources. Sophisticated software programs are used for big data analytics, but the unstructured data used in big data analytics may not be well suited to conventional data warehouses. Big data's high processing requirements may also make traditional data warehousing a poor fit. As a result, newer, bigger data analytics environments and technologies have emerged, including Hadoop, MapReduce and NoSQL databases. These technologies make up an open-source software framework that's used to process huge data sets over clustered systems.

C .Deep Analytics

Deep analytics is a process applied in data mining that analyzes, extracts and organizes large amounts of data in a form that is acceptable, useful and beneficial for an organization, individual or analytics software application. Deep analytics retrieves targeted information from data stores through data processing methodologies. Deep analytics generally extracts information from data sets that are hosted on a complex and distributed architecture, with the implementation of data analysis algorithms and techniques. The deep analytics process requires operation on a huge amount of data, typically in petabytes and exabytes. The data analysis workflow is spread out across a number of server or computing nodes to speed up the process. Deep analytics is often coupled with or part of business intelligence or data mining applications, which apply query-based search mechanisms to data stores to analyze and extract the best data match, and convert that information into specialized reports, charts and graphs.

II PROPOSED SYSTEM

The Map Reduce framework is modelled using a novel technique named as Infrequent Principle component Analysis based Classification algorithm. The proposed technique is capable of classifying and indexing the large volume and high dimensional data. It works using data reduction methods like Singular value decomposition technique to eliminate the unwanted or less relevant attributes or fields in the data Matrix taken distance computation. The Infrequent PCA is capable of partitioning and extracting the feature for data labelling of classes. Usually class contains the set of relevant features. The Random Selection strategy generates a set of samples, then calculates the pair wise distance of the points in the

sample, and the sample with the biggest summation of distances is chosen as set of pivots. It provides good results if the sample is large enough to maximize the chance of selecting points from different clusters. The Furthest Selection strategy randomly chooses the first pivot, and calculates the furthest point to this chosen pivot as the second pivot, and so on until having the desired number of pivots. This strategy ensures that the distance between each selected point is as large as possible, but it is more complex to process than the random selection method. Load balancing is also managed by efficient configuring of the proposed algorithm with map reduce function.

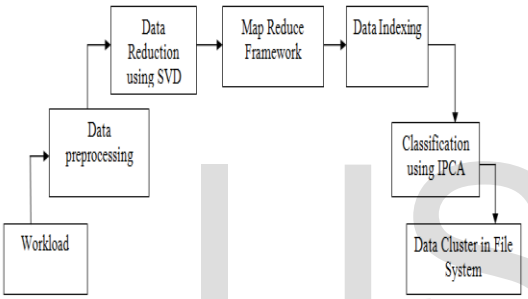


Fig.1. Architecture Diagram of our Model

Advantage of proposed System

1) Data has been classified with high accuracy.
2) Computational complexity has been gradually reduced with map reduce and SVD implementation
3) Data classification can be carried for high dimensional growing data.

Its predicts the distance for sparse data in the high dimensional space for clustering

A. Data Pre-processing

The data transformation is carried out initially .It is used to reduce the data space from stop words such as "is , was, and , the " and removing the "ing" and "ed" terms in the words using the Stop word removal process and stemming process. In this phase in which both the normalization and training of dataset is performed. Data mining approaches we need to normalize the inputs; otherwise the network will be ill conditioned. In essence, normalization is done to have the same range of values for each input to the KNN Model This can guarantee stable convergence of weights. In the training process the correct output for each input record is known and the output nodes are assigned with these correct values. The network's calculated values for the output nodes is compared to these "correct" values, and calculate an error term for each node .These error terms are then used to adjust the weights in the hidden layers so that during next iteration the output values will be closer to the "correct" values

B. Data partitioning and feature Selection

The data is partitioned into groups or partition using following strategies, such as the feature with inconsistent data or more missing value has to be considered as non actionable attribute. The reduced feature has to abide of the cases with aggregated space with retaining the important features; application of the process performance of the system is improved the features reduced on the one of the two dimensional attribute which has high correlations with similar information. The Correlation of the Attribute is calculated based on the similarity and distance function using correlation coefficient. The Coefficient measures the correlation between pairs of columns to remove one of two highly correlated data columns.

C. Modelling the map reduce Framework utilizing KNN classification

The KNN is modelled to compute the distance between the two columns in the vector space model in order to compute the similarity. After computing the distance, sorting has to be made in ascending order to extract the results. In this section, we review the different strategies used to finally compute and sort distances

efficiently using MapReduce. These different strategies can be divided into two categories, depending on the number of jobs they require for load balancing.

Algorithm

Input

Initialize the cluster centre ()
Form clusters () based on cluster centre
For all data points

Process

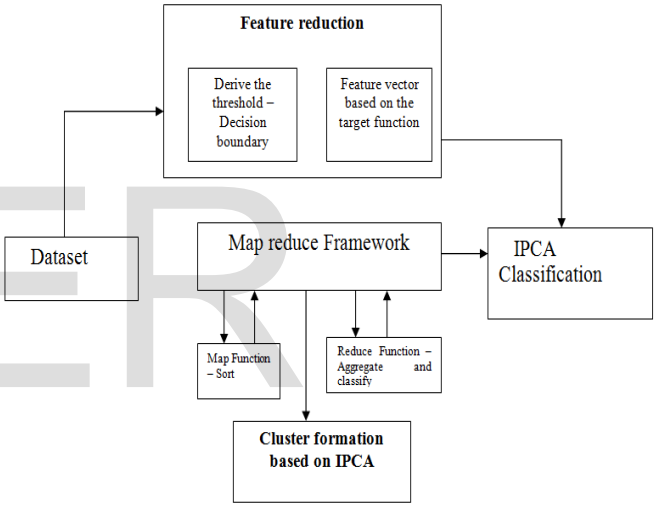
Take training data set <ds, f(ds)>
Take testing data set <ds, f(ds)>
Set N to sample values
Assign the learning rate LER
Assign N value for cross validation
Attribute valuesNormalizeby standard deviation
Set feature weights for each cluster
Normalize the features
Set random Values viFor each attribute ATi
Divide the training examples into N sets
By cross validationTrain the Values
For Eachpair Ek in N, do
Assign Ek = Validation pair
For eachsample xi in N such that xi does not belong to Ek do
Based on the Euclidean distance find the K nearest neighbors
Return the classvalues that represents the maximum of the k instances
If actual class(ac)Not Equal to predicted class(pc) then apply gradient descent
Err = no of partitions Predicted(P) Class
For each vk
vk = vk + LER * Err * Ek (where Ek is the query attribute value)
Calculate the accuracy as
Accuracy = (No.of of correctly classified samples / No.of of testing samples) X 100

Output –

Then form cluster based on the features weights

D. Modelling the map Reduce Framework utilizing the IPCA classification

The IPCA is computed to estimate the distance between the two columns in the vector space model in terms of variance. It is carried out transferring the matrix into the Eigen matrix. It uses the Covariance matrix and correlation matrix for similarity computation. It is optimally describe the cross-covariance between two documents easily. The load balancing measures with less weight in the map reduce paradigm.



Algorithm

Set all feature are Real values Numeric eg: 0,1,2
Input: Training Data (ds)
Input: Testing Data Set (ds)
N=Sample Values;
LER=Learning Rate;
N=Cross Validation;

For Each Attribute (ATi) Normalized

1) Divide Training Examples into N
2) Values Train by cross Validation
   For Each(Pair EK in N)
   Ek= Validation Pair

For Each xi !-→Ek
K=Based on Euclidean distance
Return K;
IF(!ac=pc)
Apply gradient descent;
Err=ASP;
For Each Vk
vk = vk + LER * Err * Ek;

Accuracy = (No.of of correctly classified samples / No.of of testing samples) X 100

End For

The algorithms use preprocessing and partitioning techniques to reduce this number as much as possible. The goal is to reduce the amount of data transmitted and the computational cost using IPCA. Communication overhead, which can be considered as the amount of data transmitted over the network during the shuffle phases.

- Computation overhead, which is mainly composed of two parts: 1). computing the distances,
- Finding the k smallest distances. It is also impacted by the complexity (dimension) of the data.

III RESULTS & DISCUSSION

A.Dataset Description –PRISM Dataset

The material data that has been collected from the Engineering services Concern, The raw data is provided as text files containing four attributes: observation time, material name, site id, and storage ID, Each text file contains material data for that particular unit.

B.Impact Analysis On The Data Volume

Each algorithm produces intermediate data so we compute a metric called Space requirement based on the size of existing cluster data and proposed cluster data.

Space Requirement = (Size of KNN Classified Cluster data + Size of IPCA Classified Cluster)/ Size of KNN Cluster

We measure the computing time of all algorithms for three different data input size of the dataset. As expected, the computing time is strongly related to the number of nodes. Adding more nodes increases parallelism, reducing the overall computing time.

C.Impact Analysis On Data Dimensions

Analyze is carried out on the behaviour of these algorithms according to the dimension of data. Since some algorithms are dataset dependent (i.e the spatial distribution of data has an impact on the outcome), we need to separate data distribution from the dimension. Data of various dimensions have reduced to built specific datasets by generating uniformly distributed data to limit the impact of clustering. As dimension of data increases, the execution time is greatly reduced. Nonetheless, the clustering phase of the algorithm performs a lot of dot product operations which makes it dependent on the dimension.A closer analysis shows that all phases see their execution time increase. However, the overall time is dominated by the first phase (generation of shifted copies and partitioning) whose time complexity sharply increases with dimension. Data distribution has an impact on the recall which gets much lower than the precision for Dataset .Rank Reduce is both dependent on the dimension and distribution of data.

D.Accuracy Analysis.

Performance is computed against the following metric in proposed and existing system

1)Accuracy in terms of precision , Recall and F- Measure

1) Map and Reduce phases are in parallel, but the optimal number of tasks is difficult to find. Given a number of partitions, the the number of pivots has to determined as it also important due it produces the impact the number of partitions. To improve the recall, the the IPCA is used to create duplicates in the original dataset by shifting data. This greatly increases the amount of data to process and has a significant impact on the execution time.
2)If the Size of the partition increases, the recall value of the cluster will be increased.
3)If increasing the indexes of the data, it increases the precision and recall in parallel.
Table 1- Performance analysis of the Proposed Models against Data management

Classification technique	Precision	Recall	Fmeasure	Cluster size
K-NN Classification	0.8	0.2	0.42	4
Infrequent principle component analysis	0.7	0.3	0.32	2

The Performance of the proposed and existing System is described in the Table 1. The precision, Recall and f measures compute the quality of the data clustering in the file system using optimized classification models.

Metrics used to compute

True class A (TA) - correctly classified into class A

False class A (FA) - incorrectly classified into class A

True class B (TB) - correctly classified into class B

False class B (FB) - incorrectly classified into class B

Accuracy = (TA + TB) / (TA + TB + FA + FB)

Precision = TA / (TA + FA)

Recall = TA / (TA + FB)

F-measure = 2 * ((precision * recall)/ (precision + recall))

CONCLUSION

We analysed and implemented infrequent principle component analysis to solve the computing intensive task in the high dimensional data. The major problem of data reduction and partition has pointed out through optimization of map reduce framework using the proposed Algorithm. The cost of repartitioning is currently prohibitive so, for dynamic queries, better approaches might rely on properties of scaling properties. The efficiency of these methods on data stream has been evaluated in terms of precision, recall and f measures. Moreover, we have performed a fine analysis, outlining, for each algorithm, the importance and difficulty of fine tuning some parameters to obtain the best performance.

References

[1] X. Bai, R. Guerraoui, A.-M. Kermarrec, and V. Leroy, "Collaborative personalized top-k processing," ACM Trans. Database Syst., 2011.

[2] H. V. Jagadish, B. C. Ooi, K.-L. Tan, C. Yu, and R. Zhang, "idistance: An adaptive b+-tree based indexing method for nearest neighbor search," ACM Trans. Database Syst., vol. 30, no. 2, pp. 364–397, 2005.

[3] C. Yu, R. Zhang, Y. Huang, and H. Xiong, "High-dimensional knn joins with incremental updates," GeoInformatica, 2010.

[4] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," Commun. ACM, 2008.

[5] G. Song, J. Rochas, F. Huet, and F. Magoul`es, "Solutions for Processing K Nearest Neighbor Joins for Massive Data on MapReduce," in 23rd Euromicro International Conference on Parallel, Distributed and Network-based Processing, Turku, Finland, Mar. 2015.

[6] H. V. Jagadish, B. C. Ooi, K.-L. Tan, C. Yu, and R. Zhang, "idistance: An adaptive b+-tree based indexing method for nearest neighbor search," ACM Trans. Database Syst., 2005.

[7] C. Yu, R. Zhang, Y. Huang, and H. Xiong, "High-dimensional knn joins with incremental updates," Geoinformatica, 2010.

[8] M. I. Andreica and N. T. pus, "Sequential and mapreduce-based algorithms for constructing an in-place multidimensional quadtree index for answering fixed-radius nearest neighbor queries," 2013.

[9] B. Yao, F. Li, and P. Kumar, "K nearest neighbor queries and knn-joins in large relational databases (almost) for free," in Data Engineering, 2010.

[10] D. Novak and P. Zezula, "M-chord: A scalable distributed similarity search structure," in Scalable Information Systems, 2006.

[11] A. Stupar, S. Michel, and R. Schenkel, "Rankreduce – processing k-nearest neighbor queries on top of mapreduce," in In LSDS-IR, 2010.

[12] W. Lu, Y. Shen, S. Chen, and B. C. Ooi, "Efficient processing of k nearest neighbor joins using mapreduce," Proc. VLDB Endow., 2012.

[13] C. Zhang, F. Li, and J. Jestes, "Efficient parallel knn joins for large data in mapreduce," in Extending Database Technology, 2012.

[14] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Localitysensitive hashing scheme based on p-stable distributions," in Symposium on Computational Geometry, 2004.

[15] S. Ewen, K. Tzoumas, M. Kaufmann, and V. Markl, "Spinning fast iterative data flows," in Proc. VLDB Endowment, 2012, vol. 5, no. 11, pp. 1268–1279.

[16] D. Logothetis, C. Olston, B. Reed, K. C. Webb, and K. Yocum, "Stateful bulk processing for incremental analytics," in Proc. 1st ACM Symp. Cloud Comput., 2010, pp. 51–

[17] A.Suresh (2014), "Privilege based Attribute Encryption System for Secure and Reliable Data Sharing", International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE), ISSN(Online):2320-9801, ISSN(Print): 2320- 9798, *Vol. 2, No.5, May 2014,* pp. 4099 – 4102.