Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 93 (2016) 292 – 298

Procedia Computer Science

6th International Conference On Advances In Computing & Communications, ICACC 2016, 6-8 September 2016, Cochin, India

Scalable Information Gain variant on Spark Cluster for Rapid Quantification of Microarray

Ransingh Biswajit Ray[a,*], Mukesh Kumar[a], Anand Tirkey[a], Santanu Kumar Rath[a]

[a] *Department of Computer Science & engg. National Institute of Technology Rourkela, Odisha, India 769008*

Abstract

Microarray technology is one of the emerging technologies in the field of genetic research, which many researchers often use to monitor expression levels of genes in a given organism. Microarray experiments have wide range of applications in health care sector. The colossal amount of raw gene expression data often leads to computational and analytical challenges including feature selection and classification of the dataset into correct group or class. In this paper, mutual information feature selection method based on spark framework (sf-MIFS) is proposed to determine the pertinent features. After completion of feature selection process, various classifiers i.e., Logistic Regression (sf-LoR) and Naive Bayes (sf-NB) based on Spark framework has been applied to classify the microarray datasets. A detailed comparative analysis in terms of execution time and accuracy is enumerated on the proposed feature selection and classifier methodologies, based on Spark framework and conventional system respectively.

Peer-review under responsibility of the Organizing Committee of ICACC 2016

Keywords: Big data; Hadoop; Spark; Microarray; Resilient Distributed Dataset; sf-NB; sf-MIFS; sf-LoR

1. Introduction

Microarray innovation has helped the researchers to quantify expression level of large number of genes in a single experiment. The multi-step, information intensive nature of this innovation has made an uncommon informatics and expository challenge. The 'curse of dimensionality' problem is the major obstacle in microarray data analysis, which often leads to compuational uncertainity or instability[1]. Therefore, selection of pertinent features (genes) continues to remain as a core task in analysis of diseases like cancer.

Big Data applications are gradually gaining focus, on account of the immense augmentation of information and storage that has occurred in the most recent decade. Traditional data mining techniques are facing huge computational and analytical challenges while handling these huge data. Reducing the execution time is extremely important for Big Data, which has a high computational resource demand on memory and CPU time. Distributed computing is a concept, where informations are processed independently in a distributed environment by using various parallel processing

* Corresponding author. Tel.: +91-898-451-7032
E-mail address: ransingh.b.ray@gmail.com

Peer-review under responsibility of the Organizing Committee of ICACC 2016
doi:10.1016/j.procs.2016.07.213

paradigm[2,3,4,5]. This idea has been embraced to achieve rapid processing and solve Big Data issue. Hadoop, which is based on MapReduce framework, has been exceptionally instrumental in executing high dimensional data applications on commodity clusters. While this framework is valuable for a wide range of applications, there are few which cannot be expressed productively as non-cyclic data flows. This includes numerous iterative machine learning algorithms, and also interactive data analysis tools. Spark is a new cluster computing frame-work, which bolsters applications with working sets while giving similar scalability and fault tolerance properties as MapReduce, and can outperform Hadoop by running 10-100 times faster.

In this paper Mutual Information feature selection method based on Spark framework (sf-MIFS) has been designed to select pertinent features from large microarray datasets. The classification of these datasets are performed using classifiers viz. Logistic Regression and Naive Bayes based on same Spark framework. The performance and efficiency of the proposed algorithms has been evaluated using Spark cluster with three worker nodes and one driver node.

2. Proposed Methodology

Large number of insignificant features reduces the analysis aspect of diseases like 'cancer'. This issue could be resolved by analyzing only the significant or relevant features from vast microarray data. The proposed methodology comprises of three stages:

- Missing data imputation and normalization methods, which are used for preprocessing of input data.
- MIFS based on Spark framework has been used for selecting relevant features.
- After selecting the pertinent features, classifiers, i.e., Logistic Regression (sf-LoR), Naive Bayes (sf-NB) based on Spark framework has been applied to classify (binary/multi-class) microarray dataset.

3. Basic concepts of Spark

3.1. Spark Architecture

SparkContext object defined in main program (called the driver program) is responsible for coordinating the independent sets of processes running on a Spark cluster[6]. The SparkContext can associate to any of the several cluster managers (Yarn, Mesos or Sparks own standalone cluster) that are responsible for allocating resources across applications. Once the connection is established Spark tries to acquire executors on nodes in cluster that computes processes and stores data for Spark applications. The application code (defined by Python or JAR files passed to SparkContext) is then sent to executors. Finally, SparkContext sends tasks to the executors to run.

3.2. Resilient Distributed Dataset (RDD)

RDD is a read-only distributed collection of objects partitioned across multiple machines that can be rebuilt in case a partition is lost. Elements of an RDD do not reside in reliable or physical storage; instead, Spark computes them only in a lazy fashion that is, the first time they are used in an action. This means that RDDs are by default reconstructed each time, when an action operation is performed on them. It is the core concept in Spark framework, which is a fault-tolerant collection of elements that can be operated on in parallel. RDDs are also immutable i.e., it can be modified with a transformation. Each transformation operation on RDD returns a new RDD without changing the original RDD. Transformation and Action are the two types of operations supported by RDD[7].

4. Implementation

4.1. Feature Selection Methodology

The input file, which is stored in Hadoop Distributed File System (HDFS) is a matrix of the form $N \times M$, where M and N are number of samples and features in the dataset respectively. The algorithm is divided into two sections,