A New Approach for Feature Subset Selection based on Hadoop

Ramya P V
M.Tech. Student Dept. of Computer Science Engineering
BTL Institute of Technology & Management
Bangalore,India

Shashikala B
Assistant Professor Dept. of Computer Science Engineering
BTL Institute of Technology & Management
Bangalore,India

***Abstract* - Feature Subset Selection is the procedure of choosing a subset of relevant features for use in model building as the data holds many redundant and irrelevant features. Redundant features are those which render no more information than the currently selected features, and irrelevant features provide no practicable information in any context. There are techniques for carrying out feature selection which lets in Exhaustive, Best fit, simulated annealing, Genetic Algorithm, Greedy forward selection and many other methods. Genetic Algorithms (GAs) are powerful metaheuristic techniques largely used in many real-world applications. It is a search heuristic that mimics the process of natural selection. This heuristic is routinely exploited to generate useful results for optimization and searchproblems. The consecutive performance of GAs demands significant computational ability both in time and resources. Even though some sequential models for GAs already exist, there is no framework affirming the growth of GA applications that can be accomplished in parallel. Therefore, this problem can be figured out utilizing Hadoop. Apache Hadoop is one of the common services that can be applied for parallel applications. Apache Hadoop is an open-source software framework for distributed storage part and processing of Big Data on clusters of commodity hardware. Its Hadoop Distributed File System (HDFS) divides files into large blocks and distributes the blocks between the nodes in the cluster. The paper sharpens on depicting an approach for Feature Subset Selection which is enforced on Hadoop platform using MapReduce paradigm.**

Key Words – Genetic Algorithms, Hadoop, MapReduce.

I. INTRODUCTION

The Feature subset selection (FSS) plays a significant role in the areas of data mining and machine learning. A good FSS algorithm can efficaciouslymove out irrelevant and redundant features by taking into account the feature interaction. A feature selection algorithm can be examined as the combination of a search technique for aiming novel feature subsets, along with an evaluation standard which scores the different feature subsets. The simplest algorithm is to try out each potential subset of features discovering the one which minimizes the error rate. The primary role of the model is to permit the user to focus on the views of GAs that are particular to the problem. The training dataset is fed as the input which comprises of number of records, also named as instances. As the training dataset is fed, the next step is to build a classifier. The potential of the classifier is assessed by the accuracy of the classification of the new data. Various approaches exist for subset selection which includes Genetic Algorithms. Metaheuristic techniques, such as Genetic Algorithms (GAs),make up the best alternative to obtain near-optimal solutions forsuch problems within a reasonable execution time and restricted resources. Genetic Algorithms (GAs) are a class of evolutionary algorithms, which are widely used in many domains, such as scheduling, chemistry, and biology. In particular, GAs is used to search for a nearly optimal solution in complex spaces with a computationally intensive solution. The GArepeats by performing some common operations on the picked out individuals, which are crossover and mutation. The population tends to amend its individuals by holding the strongest individuals and rejecting the weakest ones. Thefitness function should correspond to objective function of the original problem and this will be applied to judge the individuals. Generally the individuals with good fitness will beselected for the following generation. The available platform is Apache Hadoop and its easy installation and maintainability are two central aspects that lead to its great popularity.Hadoop is a generic processing framework projected to carry out queries and other batch read operations versus massive datasets that can be tens or hundreds of terabytes and even petabytes in size.Nowadays it is common for an industry torent a cluster on-line in order to request the execution of their applications as services. MapReduce is a programming model whose source lies in the old functional programming.MapReduce is a batch query processor and the entire dataset is processed for each query. It is a linearly scalable programming model where users programs at least two functions the “map” function and “reduction” functions. These functions process the data in terms of key/value pairs which are unaware of the size of the data or the cloud that they are running on, so they can be used unaltered either for a small dataset or for a massive one.

II. RELATED WORK

The Feature subset selection processed on Hadoop platform will mend the performance when equated to sequential version. The Hadoop platform processes the

prominent clusters of data and is also utilized for distributed computing. A typical MapReduce program begins on a single machine applying a piece of code called driver. This launches and supervises the execution called job of the entire distributed program on the cluster.

The article discusses how GAs can overcome many problems encountered by traditional search techniques such as the gradient based methods. The proposed controlling algorithm allows four-neighbor movements, so that path-planning can adapt with complicated search spaces with low complexities.Optimization techniques are search methods, where the goal is to find a solution to an optimization problem, such that a given quantity is optimized, possibly subject to a set of constraints [1].

The bookArtificial Intelligence: A Modern Approach helps harness the power of data. Ideal for processing large datasets, the Apache Hadoop framework is an open source implementation of the MapReduce algorithm upon which Google built its empire. This comprehensive resource demonstrates how to use Hadoop to build reliable, scalable, distributed system: programmers will find details for analyzing large datasets and administrators will learn how to set up and run Hadoop clusters[2].

The paper by J.Dean and S.Ghemawat shows the model which is easy to use, since it hides the details of parallelization, fault-tolerance, locality optimization, and load balancing. They have restricted the programming model which is easy to parallelize and distribute computations in order to make them fault-tolerant. A number of optimizations in the system are therefore targeted at reducing the amount of data sent across the network: the locality optimization allows reading the data from local disks, and writing a single copy of the intermediate data to local disk saves network bandwidth [3].

The work by C.Jin is an extension, by adding a second Reducer, of MapReduce which is named “MRPGA” based on .Net. In this, a coordinator client manages the executions of the parallel GA iterations. The chosen model is the island model in which each participating node computes GAs operations for a portion of the entire population. The Parallel Genetic Algorithms can benefit from the MapReduce model on handling heterogeneity and failures.But there are various ways of easily parallelizing other population based solutions[4].

A.Verma has proposed scaling Genetic Algorithms using MapReduce. In this paper the number of Mapper nodes and the one of the Reducer nodes are unrelated. The Mapper nodes compute the fitness function and the Reducer doesa local selection followed by the other GAs functions. The convergence and scalability of the implementation has been investigated. Adding more resources would enable to solve even larger problems without any changes in the algorithm implementation. [5].

Linda Di Geronimo has proposed a Parallel Genetic Algorithm for the automatic generation of test suites. The solution is based on Hadoop MapReduce since it is well supported to work also in the cloud and on graphic cards, thus being an ideal candidate for high scalable parallelization of Genetic Algorithms. A preliminary analysis of the proposal was carried out aiming to evaluate the speed-up with respect to the sequential execution. The analysis was based on a real world open source library [7].

The main contribution of this paper is to take into account the user to focus on views of GA that are particular to the problem, being certain that the job is going to be correctly accomplished on the hadoop cluster with a dependable performance.

III. EXISTING SYSTEM

This section keys out the methods applied for parallelizing the genetic algorithms on Hadoop and also the grain parallelism models which are carried out by exploiting the MapReduce paradigm.

The existing system exploits the Coarse-grained Parallelisation Model in which the population is separated into islands and the GA is independently run on each of them. Periodically the islands substitute the information by migrating some individuals.

The training dataset consists of records, also called instances. Each record is a list of attributes (features), in which one attribute is the class attribute. The classifier is built employing the training dataset. For instance, the C4.5 algorithm is employed in order to construct a Decision Tree, which is able to afford a probable class of ownership for every record in the new dataset.The driver is the important component of the algorithm and is carried out on one machine. There is also the possibility of assigning the accuracy target in order to terminate the process aheadachieving themaximum number of generations. Generally figuring out the accuracy for all the possible subsets involves exponential time, the idea is to present a randomly generated initial group of attribute subsets, building the initial population, to the algorithm generations. During each generation, every subset is measured by calculating the accuracy value and all the GAs functions are applied until target accuracy is achieved or maximum number of generations is reached. At the end, the last population is ready to be tested with the test dataset.

By feeding the training dataset as input, the first employed operation is the initialization. The algorithm gives the r random attribute subsets which will be the initial population of individuals. Every individual (subset) is converted as an array of m bit, where each bit shows if the corresponding enumerated attribute is present into the subset (value 1) or not (value 0). Since the records in the training dataset are never variedthroughout the algorithm, it is not essential to convert them. They will be available when demanded for the generations.

Every generation phase treats all the individuals amongst the population, according to the following steps. The fitness measure is calculated by choosing the current

portion of the dataset that is going to behave as training dataset and the part of it as test dataset. The next step is to build the decision tree through the C4.5 algorithm and calculate the accuracy by submitting the current dataset. The best accuracy is returned.

Selection process selects which individuals will be the parents during the crossover. It is important to give the individuals with the best accuracythe best probability to be chosen. The algorithm uses the method of the roulette-wheel selection, it builds a wheel according to the fitness values (accuracy) of each individual, after which it will turn for every new couple in such a way as to choose who will form it.

In the crossover phase the new offspring is produced dividing the parents of each couple into two parts according to a random crossover point. According to the probability to mutate, each subset may change the attribute to itself. The elitism step allows choosing the best individuals among the ones in the new offspring.

IV. PROPOSED SYSTEM

This section describes the method for selecting the relevant attributes which contains useful information. The initial population is selected randomly and the combination of the random attributes is taken for the fitness calculation. The fitness is calculated by using the C4.5 algorithm which builds the decision tree and also finds the accuracy for the selected attributes. The individuals with best accuracy are returned for selection. The selection process uses the tournament selection which chooses the individuals that act as parents during the crossover. Tournament selection involves running several tournaments among a few individuals chosen at random from the population.

The existing system uses the roulette wheel selection which potentially selects useful solutions for recombination. In this method, the fitness function assigns fitness to possible solutions. This fitness level is used to associate a probability of selection with each individual chromosome. The method roulette wheel selection takes O(log n) time to choose an individual. However the tournament selection has many benefits: it is efficient to code, works on parallel architectures and allows the selection pressure to be easily adjusted.

Crossover is a process of taking more than one parent solutions and producing a child solution from them. A single crossover point is done in which the data beyond the point will be swapped between the two parent solutions. The resulting solutions will be the children.The next step is the mutation which defines the probability to mutate in which each subset may change the attributes into itself. The termination criteria is defined by the user and checks whether the maximum number of generations has been reached. Here the dataset taken is the German credit dataset which contains 21 attributes. The individuals are represented in the form of numbers which indicates the attribute names.

A) System Architecture The architecture diagram shows the Initializer which takes the input and the input is split using the Splitter function which divides the input to files. The system architecture is as shown in the figure below which gives the details of every step and the MapReduce jobs applied on the steps are: i)Initializer

The initializer takes the population if it's already available or the user defines how to generate the population. The initial population consists of the attributes that are randomly selected. The combination of the randomly selected attributes serves as the initial set of population. The output of initializer is a sequence of files coming out of a MapReduce job.

ii)Splitter

The splitter processes the population and splits the individuals one per each individual according to the order. Each split contains a list of records,

iii)Fitness

The fitness value is calculated for each individual, depending on the user defined function and the values are stored inside the corresponding field of the objects. The algorithm used for calculating the fitness function is the J48 which is the open source implementation of Weka tool. J48 is a program that is written in java which builds the decision tree and calculates the accuracy for the randomly selected attributes.iv)TerminationCheck

This checks whether the current individual satisfies the user the termination criterion. The termination criterion is defined by

v)Selection

If the termination criterion is not satisfied, it then chooses the individuals that will be the parents for the next iteration. The tournament selection is the method that is deployed in order to select the best individuals. Tournament selection involves running several tournaments among a few individuals chosen at random from the population. The winner of each tournament is selected for crossover. Selection pressure is easily adjusted by changing the tournament size. If the tournament size is larger, weak individuals have a smaller chance to be selected.

vi)Crossover

The individuals are grouped in this phase by the couples established during the selection. This step produces the offspring together with the previous population.

vii)Mutation

Here the mutation is performed only on the offsprings. A mutation rate that is too high may lead to loss of good solutions. The chained mappers manage to make the mutation of the genes defined by the user.

viii)Elitism

The elitism is optional and if the user chooses to use elitism the definitive population is chosen among the individuals of the offspring and the previous population. The selection of elitism guarantees that the solution quality obtained by GA will not decrease from one generation to the next.

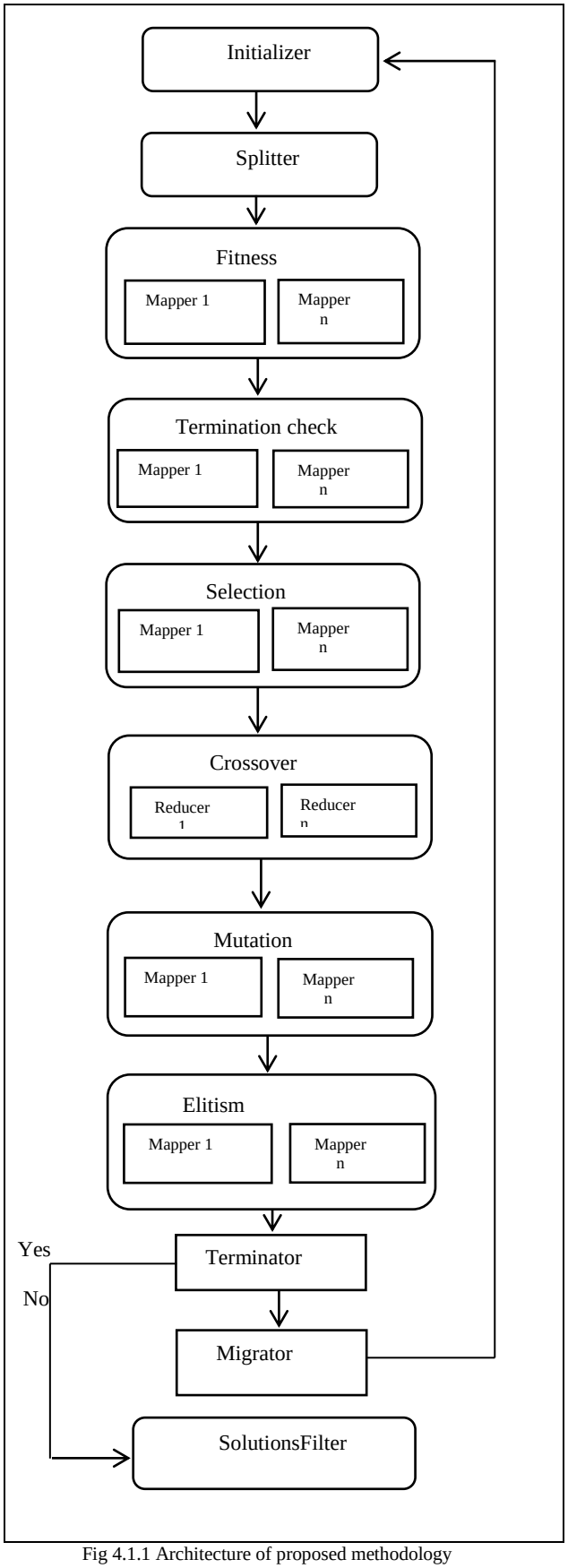

Fig 4.1.1 Architecture of proposed methodology

B. Modules

1) Construct Genetic Algorithm functions:

The GA functions are initializing the population; calculate the fitness, selection, crossover, mutation and elitism. For the Initialization process, the algorithm generates the random attribute subsets which serve as the initial population. The fitness is calculated for the random attributes by using the C4.5 algorithm which builds the decision tree and gives the accuracy for the random attributes. After calculating accuracy for the random attributes, the individuals with best accuracy are returned. Next step is the selection process; it chooses which individuals will be the parents during the crossover. The algorithm used for selection is the tournament selection which selects the individuals from a population of individuals. During the crossover stage, the new offspring is produced by splitting the parents of each couple into two parts, according to the random crossover point, and then mixing the parts obtaining two children. In the mutation step, each subset may change the attributes into itself. The elitism step is optional, which allows choosing the best individuals among the ones in the new offspring and in the previous generation, in such a way as to guarantee the growth of the accuracy target after every generation.

2) Applying the GA functions on Hadoop platform:

The random attribute subsets are given to the Initialiser as the initial population. The next step is to split the individuals into group islands according to the order of individuals. The fitness function takes the input from splitter and calculates the fitness function for the attribute subsets using the C4.5 algorithm which builds decision tree according to the user defined function and also calculates the accuracy. The next step is the termination check which returns the individuals that satisfies the criterion. The individuals satisfying the criterion will be selected for the next phases such as selection, crossover and mutation. The crossover step is done in the reducer phase. The mutation and the elitism are done in the mapper phase.

3) Performance Analysis:

The GA functions are executed on sequential version and the performances are recorded. The Genetic algorithm functions are also implemented on the Hadoop single node and on the multinode environment. The performances will be compared in order to check how the Hadoop processes the large-scale data in the distributed environment.

C. Results

The results achieved are performed on the dataset German credit and audiology which are taken from the UCI Machine Learning Repository. The dataset was divided into two parts in which first 60% is taken as training dataset and the last 40% as the test dataset which is been used to test at the end. It is executed on Hadoop single node and has achieved the best accuracy. The model will be executed on multi node cluster in order to check the performances of both single and multinode cluster.

V. CONCLUSION

The feature selection on Hadoop is done in order to defeat the sequential version of Weka, which is specialized in Machine learning algorithms. The performances of the MapReduce jobs show that writing GAs with the framework is effective which also gives good performance. Since it is possible to rent a cluster for small amount without investing on an expensive and dedicated hardware, everything will be completely operative so as to run any type of distributed applications.The future implementation of this project is to analyze the possible bottlenecks caused by the inner structure of Hadoop.

REFERENCES

[1] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley LongmanPublishing Co., Inc., 1989.

[2] S. J. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, Third. Prentice Hall, 2010.

[3] J. Dean and S. Ghemawat, "Mapreduce: simplified data processing on large clusters", OSDI, 2004.

[4] C. Jin, C. Vecchiola, and R. Buyya, "Mrpga: an extension of mapreduce for parallelizing genetic algorithms", in eScience, 2008. eScience '08. IEEE Fourth InternationalConference on, IEEE, 2008, pp. 214–221.

[5] A. Verma, X. Llor`a, D. E. Goldberg, and R. H. Campbell, "Scaling genetic algorithms using mapreduce", in Intelligent Systems Design and Applications, 2009.ISDA'09. Ninth International Conference on, IEEE, 2009, pp. 13–18.

[6] S. Di Martino, F. Ferrucci, V. Maggio, and F. Sarro, "Towards migrating genetic algorithms for test data generation to the cloud", in. IGI Global, 2012, ch. 6, pp. 113–135.

[7] L. Di Geronimo, F. Ferrucci, A. Murolo, and F. Sarro,"A parallel genetic algorithm based on hadoop mapreducefor the automatic generation of junit test suites",in Software Testing, Verification and Validation (ICST), 2012 IEEE Fifth International Conference on, IEEE, 2012, pp. 785-793.