CS 5433 - Big Data Management

Spring 2018

Big Data Group Project – Phase I

Group: 3

Baath, Harpinder

Polavarapu,Tejaswanth

Shankara, Akshay

Dandamudi, Madhu Kira

Git Repository:

https://github.com/akshayshankar/big-data-group-project

2. Problem Understanding:

Feature selection is an important aspect of predictive analytics. It not only tends to reduce the complexity of an algorithm but improves performance and cost of data collection there by selecting sensitive variables which can provide significant predictive accuracy. The optimum feature subset selection is a NP hard problem which cannot be solved in polynomial time, genetic algorithms provide near optimal solution to this problem. Genetic algorithms are motivated from Darwin's theory of survival of the fittest and mimic biological reproduction process. It starts with random population of individuals (represented as chromosomes) and then evaluate the fitness of each individual. Two individuals are selected on the basis of their fitness value to serve as parents for crossover intending to produce better offsprings. Individuals with higher fitness value are more likely to be selected. Genetic algorithms need a lot of computational power and resources, fortunately, hadoop MapReduce allow parallelism which can be used to run genetic algorithms in parallel and obtain the desired result with improved efficiency.

In feature selection problem, each feature represents a gene and the collection of genes is a chromosome (subset of features). Each chromosome is represented by a string of 0s and 1s. 0; if feature is absent, otherwise 1. The population is a set of randomly selected chromosomes. With the help of the training and the test dataset obtained from the HDFS, the 'accuracy' (fitness) of each individual is calculated using machine learning algorithms such as decision tree, support vector machine etc. For each reproduction, the parents with the best fitness are selected from the pool of population using 'roulette wheel' phenomenon and a certain crossover criteria is used to produce new subsets which include features from both the parent subsets and some error (mutation). With each generation of population, we aim to achieve a better fitness and for this, we select a better set of individuals. This process repeats until the near optimal to the desired results of fitness are reached.

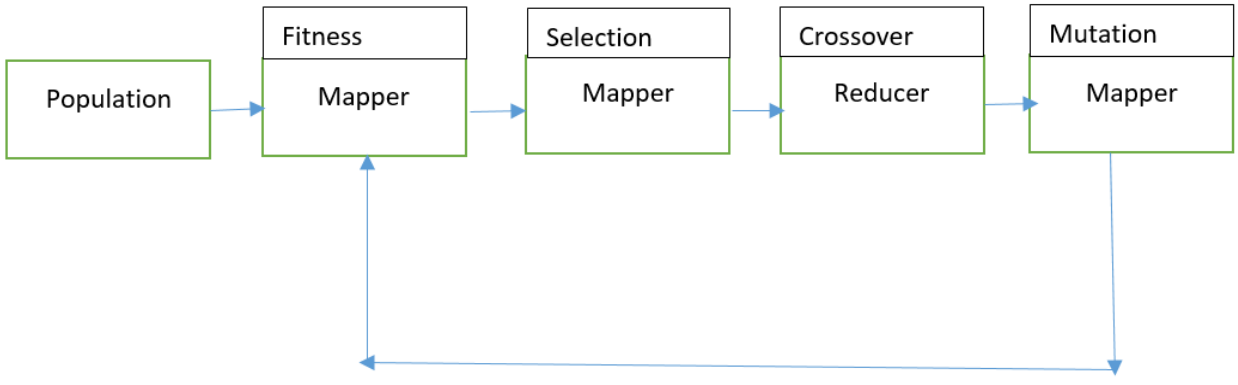

Figure: Feature subset selection process in MapReduce using GA

3. Tasks and deadline:

- Understanding how MLib library works in Spark. 03/28
- Working with feature selection with respect to the data. 03/30
- Converting feature into chromosome representation. 04/04
- Implementing machine learning algorithms in map reduce to calculate predictive accuracy. 04/06
- Implementing Selection method in Mapreduce. 04/10
- Implementing Crossover and mutation methods in Mapreduce. 04/13
- Integrating all the above methods in Mapreduce. 04/21
- Testing and debugging. 04/21 - 04/26

4. Understanding of the wrapper method and its implementation idea on the given dataset:

Genetic algorithms are well known for solving optimization problems by iteratively running biologically inspired set of operations until the desired result is obtained. These operations are:

- **Selection**: Each individual is assessed on the basis of the value of fitness function. The fitness value effects the probability of an individual to be selected as parent for producing new offsprings.
- **Crossover:** Algorithm chooses a crossover point to split both the parent individuals in halves and start reproduction process for producing two children. New children possess parts of genes from both the parents depending upon the crossover point.
- **Mutation**: This adds a little error in some of the genes of newly created offsprings.

For feature selection problem, GA starts with finding fitness value of each set of features using machine learning algorithms such as decision tree, support vector machine etc. for the given dataset. For each reproduction, the parents with the best fitness are selected from the pool of population using 'roulette wheel' phenomenon and a crossover criteria is used to produce new subsets which include features from both the parent subsets and some error (mutation). With each generation of population, we aim to achieve a better fitness and for this, we select a better set of individuals. This process repeats until the near optimal to the desired results of fitness are reached.

Various components of GA for feature selection are given below:

- **Encoding technique**

The chromosomes (features) are encoded using one of the following encoding techniques:

- Binary Encoding: The chromosome is transformed into a string of 1's and 0's, where each bit indicates the presence or absence of a value.
- Permutation Encoding: The chromosome is represented by a sequence of numbers that, usually, represents the order in which a certain process is performed.
- Value Encoding: Chromosomes are directly encoded with the data. This is used when the data format is complex.

- **Initialization procedure**

The initial population (set of features) is selected randomly and is usually in abundance, so that it contains the entire set of probable solutions.

- **Fitness function**

The fitness function determines the fitness value of a particular chromosome or the optimality of the chromosome. Fitness value for the each set of features is calculated using machine learning algorithms such as support vector machine, decision tree etc. for the given dataset.

- **Selection of parents**

Depending upon the evaluation function mentioned above, we select the best chromosomes from the available set of population. Roulette wheel selection criteria will be used to select best possible feature sets for crossover on the basis of their predictive accuracy. A feature set with more predictive accuracy would more likely to be selected.This process of selecting the best possible chromosomes is relatively termed as selection of parents. This could be implemented as below:

- Select random element from population
- Calculate its fitness
- Check if fitness matches a minimum threshold
- If fitness does not match the threshold go back to the first point
- Repeat the same process to get 2 elements from the population
- Select these two elements and term them as parents

- **Genetic operators**

Once we obtain the parents based upon the fitness function, the next thing we do is recombine them to produce the offsprings. For crossover, based on the crossover point each parent feature set would split into two and then two new offsprings would be generated by combining the features from both the parents. The first half of the first parent would combine with second half of second parent to generate the first child and vice versa for the second child.

When offsprings are produced, some amount of mutation would be added into their genes. Mutation would be kept to be very low (ex: <0.03%) because this doesn't likely contribute in producing the best offsprings and our goal might be compromised.

- **Parameter settings**

There are certain parameters that needs to be set for proper working of GA. The parameters to be taken care of are:

- Set the size of the population
- Set the rate for mutations
- Set the parameters for crossover(like setting the crossover rate and crossover point)