PPD-GAR: Parallel processing of Data with Genetic Algorithm using R

Chetana Tukkoji [1] and Dr. Seetharam K [2]

1. **Assistant Professor, Department of CSE, GITAM Deemed to be University -Bengaluru.**
2. **Professor, Department of CSE, N.M.A.M Institute of Technology- NITTE**

ABSTRACT: *A genetic algorithm (GA) gives optimal solution for the problems, which creates a group of populations and identifies possible solutions to the given problem. The individuals in this population will carry solutions that are the values of variables of the problem. The process of genetic algorithm involves the selection, crossover and mutations are called genetic operators. This paper uses RStudio platform for analyzing super sale market store dataset using genetic algorithm and also compares the different algorithm such as LDA, CART, KNN, SVN and RF parallel, to estimate accurate analysis results of dataset.*

Keyword: Parallel processing, Genetic operators, Dataset.

I. INTRODUCTION

Over the last years the Internet of Things (IoT) has gained more attention from both industry and academia., day by day generation of data is more from, wireless network, sensors, social media, scientific applications etc is called bigdata. The big data are in the form of text, audio, video, images and many more generated from different business operation. Storage of big data is not a problem as cloud offers ubiquitous services. With a huge volume of data which is being generated and achieving effective storage and data processing becomes a challenging task with respect to cost and optimization of data. Here we have considered the different algorithms for analyzing dataset related to super sale market for data portioning to estimate the accuracy of each algorithm and comparing the results for better efficiency, that in turn will benefits for the business operation to improve percentage of profits on each sale product. Applying genetic algorithm includes selection, crossover and mutations on given dataset for finding individual solution for each attributes and finally optimizing the solution for the given problem. This paper considers the following different algorithm for comparative study of given dataset.

A). Linear Discriminant Analysis (LDA)

LDA method is used in statistics, machine learning and pattern recognition in order to find a linear combination of the feature set that categorizes two or more types of objects. The resulting combination may be used as a linear classifier. LDA method works when the measurements made on independent variables for each observation are continuous series of quantities. While dealing with categorical independent variables, the equivalent technique is called as discriminate analysis.

B). Classification and Regression Trees (CART)

A CART is one of the commonly used methods for Decision Tree algorithms. The algorithm involves number of procedures, those are: collecting the input data and labeling them with Target Variable and a list of Independent Variables, partitioning the dataset based on each of the independent variables, and finally selecting the best fit variable for split up of data, repeat the procedure until it meet stopping criteria**,** and finally prune the decision tree which has been built.

C). K-Nearest Neighbors (KNN)

K-means is also called as K-nearest neighbors is an algorithm used to classify dataset using Euclidian distance function to similarity measure. KNN has been used in statistical estimation and also for pattern recognition.

D). Support Vector Machines (SVM)

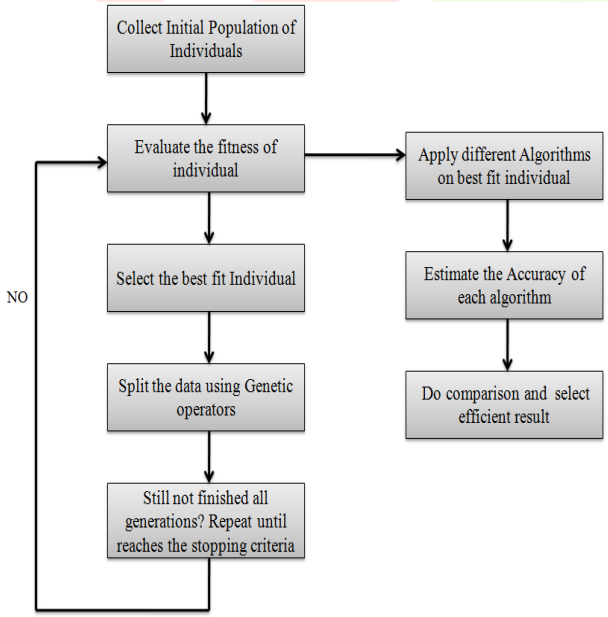
Support vector machines (It also support for vector networks) which are categorized as supervised learning models with associated learning algorithms that analyze dataset which will be used for classification and regression analysis. For example, consider a set of training dataset, each has been marked as label which belongs to one or the other of two categories, the algorithm also builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier.

E). Random Forest (RF)

Random forests (RF) is a method used for classification, regression and other tasks, that operate by constructing a large number of decision trees during training time and operating the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. This algorithm provides a good way to implement the stochastic discrimination approach to classification [5].

II. GENETIC ALGORITHM

Figure 1. Data flow diagram of genetic algorithm (GA)



The prime aim of the GA is to provide optimal solution for the individuals. The algorithm involves genetic operators. Figure 1. Show the data flow diagram of GA, where it computes values based on genetic operators. Genetic operators are used in genetic algorithms to guide the algorithm towards a solution to a given problem. Mainly there are three types of operators called mutation, crossover and selection [1].

1. **Mutation:** Mutation which alters its initial state of population and solution to this may change entirely from the previous solution. Hence GA can come to a better solution by using mutation. Usually mutation occurs during evolution. This can be used for primitive random search algorithm.
2. **Crossover:** Crossover is a process of taking producing more than one solution to the given problem with different methods such as Simplex Crossover and multiple crossover
3. **Selection:** Selection algorithms do not consider all individuals for selection, but only those with a fitness value that is higher than a given arbitrary value. Where other methods used to select from a restricted pool with certain percentage of the individuals are allowed, based on fitness value[2].

In order to find best results for partition the given dataset for parallel processing, different algorithms are applied such as LDA, CART, KNN, SVM and finally RF, and comparative results shown the best algorithm to choose for data partitioning.

III. ALGORITHM IMPLEMENTATION

In order to evaluate the dataset called ‘Super Sale Market Store ’ with 8399 of records with set of 11 attributes , in that we have considered only 7 attributes (OrderID, Sales, Profit, UnitPrice ,ShippingCost, OrderQuantity, OrderPriority) for data split up for parallel processing with different algorithms , the operation is performed and also model is simulated using RStudio, the observations of the different algorithms are made.

- **Data partitioning:**

Creating a list of 80% of the rows in the original dataset can be used for training set and for testing models. And remaining 20% of dataset can be used for data validation[4].

Now check for all each attributes and its type of class belong to using typeof() function

Table 1. List of Attributes and its class type

Attributes	OrderID	Sales	Profit	UnitPrice	ShippingCost	OrderQuantity	OrderPriority
Class	integer	integer	Numeric	numeric	numeric	integer	character

Now have a look at the first eight rows of dataset values as shown bellow.

Table 2. Sample 8X7 records of 'Super Sale Market Store'

Attributes / Class & SI. NO	OrderID <int>	Sales <int>	Profit <dbl>	UnitPrice <dbl>	ShippingCost <dbl>	OrderQuantity <int>	OrderPriority <chr>
1	2	142	4.90	3.00	2.10	8	Medium
2	3	168	4.70	3.20	1.80	6	High
3	4	457	4.60	3.10	1.80	5	low
4	5	467	5.00	3.60	1.80	5	Medium
5	6	682	5.40	3.90	2.10	4	High
6	12	1513	4.80	3.40	1.40	6	High
7	1	92	5.01	3.05	1.08	6	low
8	13	1535	4.08	3.00	2.03	7	Medium

Find out the levels of attribute type called 'OrderPriority' , and calculate the statistical summary for each attribute.Design a probability table to maintain the percentage of occurrences of class type of attribute 'OrderPriority'.

Now we can see that each class has almost same number of instances (37 or 25% of the dataset) as shown in the Table 3.

Table 3. Number of Occurrences of level type with its percentage

SI. NO	No. of Occurrences	Frequency	Percentage
1	3	High	40
2	3	Medium	40
3	2	Low	25

We can see that all of the numerical values have the same scale (in centimeters) and similar ranges [0,8] centimeters.

We need to check total summary (dataset) that gives the results of 8 X7 records with six different parameters such as min, max, 1st Qu, median, mean, 3rd Qu in the Table 4.

Table 4. Summary of dataset with six attributes (having numerical value)

Attributes / SI. NO	OrderID <int>	Sales <int>	Profit <dbl>	UnitPrice <dbl>	ShippingCost <dbl>	OrderQuantity <int>	OrderPriority <chr>
Min	2.00	47	4.300	2.00	1.400	1.000	Level=Low
Max	99.00	107700	7.000	4.400	2.500	9.000	Length=98
1st Qu	26.25	4354	5.000	2.800	1.800	4.000	Class=character
Median	50.50	75301	5.400	3.050	1.900	6.000	Mode=character
Mean	50.50	43365	5.472	3.093	1.973	5.742	[As it is having
3rd Qu	74.75	79149	5.900	3.400	2.200	7.000	character value]

- **Splitting input and output**

  Consider the values of dataset for both X and Y axis is with 6 column (numerical values) and 8 rows a respectively

  X ← dataset[,1:6]

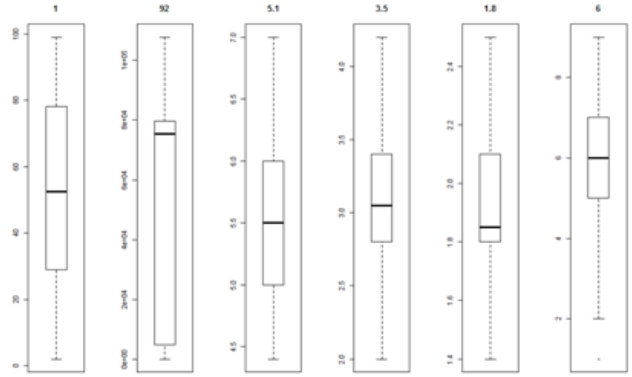
  Y← dataset[,8]

- **Generation of plotting window**

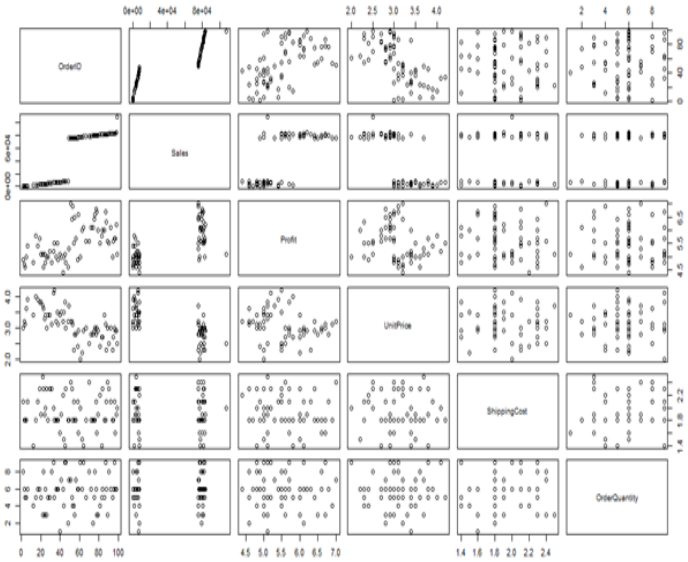
  Based on the X and Y attribute values, multi paneled plotting window is generated using par() function.

```
//boxplot for each attribute on one image
par(mfrow=c(1,7))
for(i in 1:7)
{
 boxplot(X[,i], main=names(StoreSale)[i])
}
```

Figure 2. shows the boxplot for each attribute on one image

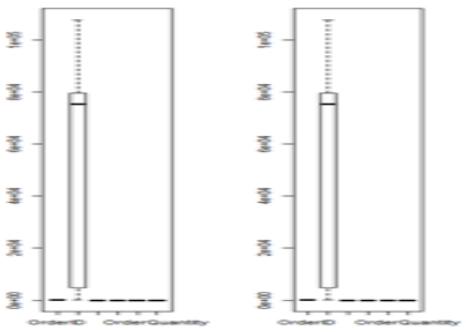


barplot for class breakdown Ex. Plot(X)
Figure3. Shows the boxplot for class breakdown



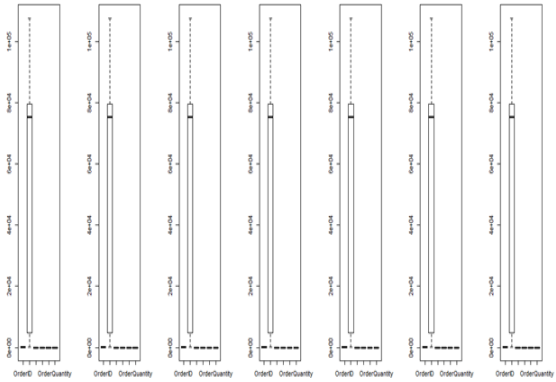
# box and whisker plots for each attribute
featurePlot(x=x, y=y, plot=boxplot(x, y))

Figure 4. shows the featureplot of X and Y



# Density plots for each attribute by class value
scales <- list(x=list(relation="free"), y=list(relation="free"), z=list(relation="free"))
featurePlot(x=x, y=y, z=z,plot=boxplot(x,y), scales=scales)

Figure 5. Density plots for each attribute by class value



- **Model Building**

As we cannot decide which algorithm will be good on this problem or what configurations to use. Generating plots are necessary.

- **Evaluation of five algorithms**

Run algorithms using 10-fold cross validation
control ← trainControl(method="cv", number=10)
metric ← "Accuracy"

This is a mixture of simple linear (LDA), nonlinear (CART, KNN) and complex nonlinear methods (SVM, RF). We reset the random number seed before each run of algorithm is performed using exactly the same data splits. It ensures the results are directly comparable, and selecting the best model.

We can report on the accuracy of each model by first creating a list of the created models and using the summary function for comparing results.
#summarizing the accuracy of models
results <- resamples(list(lda=fit.lda, cart=fit.cart, knn=fit.knn, svm=fit.svm, rf=fit.rf)) and summary(results)

We can see the accuracy of each classifier and also other metrics like Kappa; We can also create a plot of the model evaluation results and compare the spread and the mean accuracy of each model. There is a population of accuracy measures for each algorithm because each algorithm was evaluated 10 times (called 10 fold cross validation).
Table 5. Summary of Metrics such as Accuracy and Kappa

Models: lda, cart, knn, svm, rf
Number of resample: 10
Accuracy

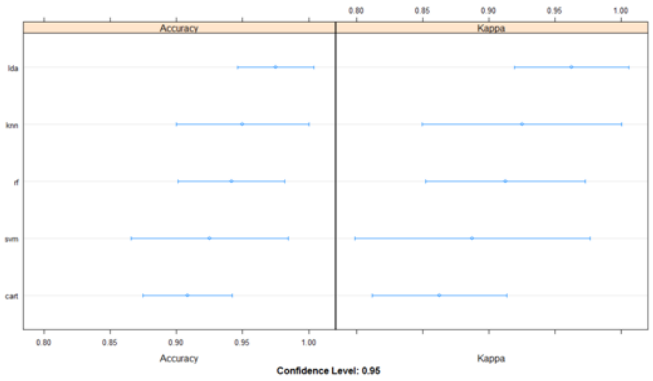
	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
lda	0.9167	0.9375	1.0000	0.9750	1	1	0
cart	0.8333	0.9167	0.9167	0.9417	1	1	0
knn	0.8333	0.9167	1.0000	0.9583	1	1	0
svm	0.8333	0.9167	0.9167	0.9417	1	1	0
rf	0.8333	0.9167	0.9583	0.9500	1	1	0

Kappa

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
lda	0.875	0.9062	1.0000	0.9625	1	1	0
cart	0.750	0.8750	0.8750	0.9125	1	1	0
knn	0.750	0.8750	1.0000	0.9375	1	1	0
svm	0.750	0.8750	0.8750	0.9125	1	1	0
rf	0.750	0.8750	0.9375	0.9250	1	1	0

Based on the plotting window result, we can see that the most accurate model in this case is LDA

Figure 6: Comparison of five algorithms using metrics such as Accuracy and Kappa



- Identifying the best model using print() function for each algorithm.

#Summarize Best Model using
print(fit.lda), print(fit.cart), print(fit.rf), print(fit.svm), print(fit.knn).

The above function gives a nice summary of what was used to train the model and the mean and standard deviation (SD) accuracy achieved, specifically 96.7% accuracy +/- 4%. Among five algorithms, LDA gives best results where it nearly gives 100% Accuracy as well as in Kappa Metrics.

Linear Discriminant Analysis

99 samples
4 predictor
3 classes: 'low', 'medium', 'High'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 90, 90, 90, 90, 90, 90, ...
Resampling results

Accuracy	Kappa	Accuracy SD	Kappa SD
0.975	0.9625	0.04025382	0.06038074

IV. CONCLUSION

Genetic algorithms are intrinsically parallel. Many other algorithms are serial and can only inquire into the solution space to a problem in one direction at a time since GAs have different view for providing a multiple solutions in multiple directions at once. If one path takes to the dead end, that can be easily eliminated and continue work on the other way with greater chance for finding the optimal solution. Genetic algorithms provide a comprehensive search methodology for machine learning and optimization. It has been shown to be efficient and powerful through many data mining applications that use optimization, classification and regression. GAs can rapidly locate good solutions.

Genetic operators are adopted in this paper for data partitioning using RStudio, where it provides a very good platform for five different algorithm comparisons with various fields of dataset to get an optimal solution using decision making methods to estimate algorithm efficiency. There is also a greater scope of GA in data mining in future application to stimulate the data mining concepts. Genetic algorithms are widely applicable to classification by means of machine learning, where it also provides a practical method for optimization of data preparation and data transformation steps. Hence GA can be used in a real analysis system to get the better optimal solution.

ACKNOWLEDGEMENT

The author would like to thank for all the referred journals for gaining knowledge on genetic algorithm. Especially we would like to thank Dell-EMC, ICT Academy for training us to know about basics of R and made interest to learn.

REFERENCES

1. https://en.wikipedia.org/wiki/Genetic_algorithm
2. Meeta Kumar et.al “Survey on Techniques for Plant Leaf Classification” International Journal of Modern Engineering Research (IJMER), Vol.1, Issue.2, pp-538-544 ISSN: 2249-6645.
3. Mário Antunes et. Al “Vehicular dataset for road assessment conditions” 978-1-5386-2524-8/17/$31.00 © 2017 IEEE.

4. https://www.statmethods.net/r-tutorial/index.html
5. https://machinelearningmastery.com/machine-learning-in-r-step-by-step