IJRSET SEPTEMBER 2020 Volume 7, Issue 9 Pages: 1-7

ISSN 2394-739X

International Journal for Research in Science Engineering and Technology

PARALLEL DECISION TREE ALGORITHM FOR MULTITEXT CLASSIFICATION BASED ON SPARK

[1] P. TAMILSELVAN, [2] DR. S. M. JAGATHEESAN,
[1] SCHOLAR, [2] ASSOCIATE PROFESSOR,
[1, 2] PG & RESEARCH DEPARTMENT OF COMPUTER SCIENCE,
[1, 2] GOBI ARTS AND SCIENCE COLLEGE (AUTONOMOUS),
[1, 2] GOBICHETTIPALAYAM, Erode Dt., TAMILNADU 638453.

ABSTRACT: One of the most challenging issues in the big data research area is the inability to process a large volume of information in a reasonable time. Hadoop and Spark are frameworks for distributed information processing. Hadoop is a very famous and standard platform for massive facts processing. Because of the in-memory programming version, Spark as an open-supply framework is suitable for processing iterative algorithms. With the rapid growth of data amount and feature space dimension under the background of big data, the parallelization of traditional multitext classification algorithms will significantly improve its running efficiency. In this paper, Spark frameworks, the big data distributed processing platforms, are evaluated and compared in terms of Precision, Accuracy and Recall. Hence, the parallel j48 pruned decision tree classification algorithm is implemented on datasets with different sizes within Spark. The results show that the runtime of the parallel j48 pruned decision tree classification algorithm implemented on Spark is faster than Hadoop. Evaluations show that Hadoop makes use of greater sources, such as crucial processor and network. It is concluded that the Spark is more effective than Hadoop.

Keywords: [Big data, Hadoop, Spark, parallel j48 pruned decision tree classification.]

1. INTRODUCTION

The sharp growth of the amount of Internet Chinese text facts has extensively prolonged the processing time of classification on that information. With the speedy increase of information amount and function space size beneath the heritage of big information, the parallelization of traditional Chinese textual content category algorithms will significantly enhance its strolling performance. The parallel naive Bayes set of rules primarily based on Spark is used to manner large-scale texts; it shows top speed up as well as scalability. Specifically, the processing time is substantially decreased in comparison with Hadoop-based model [1].

Apriori set of rules is one of the fundamental algorithms to locate frequent item sets from transactional statistics amassed for market basket analysis. It is a famous fact that it suggests overall performance bottleneck because of several motives like the high wide variety of candidate generation in every new release, requirement of large primary reminiscence for quicker computation, and repeated scans of the original input dataset. Thereafter, many variations of the apriori set of rules have been proposed using the MapReduce surroundings to enhance the performance of the set of rules by way of making the execution parallel. For EAFIM, Apache

Spark appears to be a better framework than MapReduce for processing large datasets due to its in-memory processing skills. Apart from utilizing the Spark environment, to boom the efficiency [2].

Swarm intelligence meta-heuristic algorithms together with Whale Optimization Algorithm are actually a few of the maximum widely used algorithms used in fixing optimization issues in numerous fields. Spark WOA a disbursed implementation of Whale Optimization Algorithm on Apache Spark platform. To examine the overall performance of Spark-WOA, various experiments had been performed on different benchmarks and had been proven to have quick convergence and accordingly green exploration of the quest area. Moreover, Spark-WOA turned into compared to Hadoop-primarily based implementation where it became discovered to offer superior effects in terms of run time and scalability [3].

The boom and expansion of the extent of information has come to be a unique phenomenon. Analyzing and storing such a large quantity of facts requires new ideas that could process and control this extent of records. All fields are interested in massive data due to its first-rate capability. An evaluation of every node within the cluster in terms of reminiscence utilization, CPU utilization, and community utilization in Hadoop and Spark platforms. For this evaluation, the KNN set of rules changed into executed on the Higgs dataset on Hadoop and Spark structures and the overall performance parameters of hobby for each node have been monitored with the aid of the software Ganglia [4].

CBIR is an attractive area of research that specializes in effective feature extraction.MapReduce programming is preferred inside the recent works due to its suitability for distributed big scale records processing. Similarly on the retrieval aspect, KNN has received its popularity due to its efficiency in classifying unlabelled records. Consideration processing KNN queries on large datasets, in which the index is maintained in a computing cluster. MapReduce framework offers quicker indexing and a parallel KNN set of rules with cache method allows to retrieve photographs faster than local methods [5].

The growing hole among customers and the Big Data analytics calls for innovative gear that deal with the demanding situations confronted via big information extent, range, and pace. Therefore, it will become computationally inefficient to analyze such large quantity of statistics. Moreover, an structure that analyzes Big Data is also proposed based on the parallel algorithm this is exploited on Hadoop server giving high-performance computing. The whole machine is applied the usage of more desirable MapReduce with the additional function of parallel processing algorithms to technique huge graphs and MapReduce to procedure different facts with Hadoop atmosphere to attain the efficiency and actual-time processing[6].

The upward thrust of high-resolution and excessive-throughput sequencing technologies has driven the emergence of such new fields of application as precision medicine. However, this has also caused an growth inside the garage and processing necessities for the bioinformatics equipment, which could most effective be provided via excessive-performance and huge data processing infrastructures. Basic Local Alignment Search Tool set of rules permit the development of scalable, green and dependable bioinformatics gear [7].

A benchmark evaluating the training and assessment of records-pushed forecasting models the use of different amounts of records, as well as R and Spark on a single computer and Spark on a computing cluster is offered. Te received consequences show the points at which a Big Data computing framework based on Spark may be superb, for example, whilst the use of a complicated records mining method or when surpassing a specific quantity of facts. Te former is shown through the truth that Spark on the cluster has for the carried out benchmark the lowest computation times for training and comparing a complex records driven model [8].

Improving Spark performance with electricity saving goals can have a

widespread effect in lowering power consumption in cloud statistics facilities. An energy aware scheduling algorithm (EASAS) to lessen the energy intake in Spark cluster. Based on historical method table, EASAS allocates duties to the ideal executor with minimal energy intake with closing date constrains. Different types of workload in benchmarks including Sort, TeraSort, K-approach, and Page Rank are achieved to investigate the overall performance of EASAS in unique instances in detail. Although the execution time of EASAS has barely increased for some workloads, EASAS can drastically lessen the energy intake of the Spark cluster [9].

Fusion effect of SVM inside the Spark architecture for speech facts mining in cluster shape is studied on this manuscript. Based on the statistics entropy of nodes, the facts in clusters are fused to do away with redundant statistics and enhance the efficiency of facts fusion. Information entropy is a statistical shape based on the traits of facts illustration, which reflects the common amount of statistics in statistics. Based at the Spark platform SVM set of rules, the frequent gadgets with the best assist after every type are immediately recursively received, and the transaction facts set is allocated to every computing node. The speech records mining set of rules can cluster, analyze, and comprehensively detection the saliency statistics, the detection accuracy is tons better than the modern-day fashions [10].

The shortcomings are advanced to enhance the accuracy and processing pace of clustering effects. In the parallel adaptive Canopy-K-method clustering set of rules proposed on this paper, the optimization step by step approximation manner will use a couple of Canopy algorithm, which has a high time complexity. Based on no longer affecting the category accuracy of the set of rules, a way to similarly enhance the efficiency of the algorithm is the subsequent studies course [11].

The parallel framework and pseudo-code description of dsPSOK-method are given. The algorithm adopts the strategy of abandoning speed and adjusting the inertia weight dynamically by way of fitness price. So that the dsPSO set of rules have adaptive traits. The output of the dsPSO algorithm is used as the enter of the K-way set of rules [12].

2. EXISTING SYSTEM

Using parallel Naive Bayes text category getting to know systems examine the efficacy of the usage of Internet Chinese textual content dataset, in which an overall muscular characterization is needed based at the examiner of the hassle, with a couple of samples drawn from the equal source. Some further exploration of these ideas using research drawn from synthetically generated blanketed statistics has been additionally performed. Values for an person RDD are calculated by the usage of a easy aggregation of all of the discovered values for every function in the observe, and adding this result as a brand new function to all samples, providing every sample statistics approximately the complete have a look at. Parallel Naive Bayes Learning is relevant in specific manipulation of probabilities many of the maximum practical tactics to positive kinds of getting to know issues.

Algorithm 1: Parallelization of NB classifier's training process based on Spark

Input: the preprocessed training set

Begin

Step 1. Define ZeroCombiner [class, (text number, Conditional probability in this class of every function)] for map data structure

Step 2. Initialize the value of ZeroCombiner and calculate the total number and feature vector sum for local samples **for** each class **i do**

Calculate the total number and characteristic vector sum for global samples end for i

Step 3. Obtain class number C and total number of training samples X. Calculate the denominator of class prior probability $P(y_i)$ and take the logarithm, piLogDenom = math.log (X+C * lambda)

Step 4. **for** each class **i do**

Obtain the number of samples in every class n, calculate class prior probability, pi(i) = math.log (n + lambda) –

Output: the training model made up of matrix **theta** and vector **pi** After the training method, the NB classification version, specially represented through sparse matrix of class prior vector and class conditional probability, is obtained.

Next, the parallelization of predicting procedure primarily based on the mounted version. Firstly, the check samples are input to form the RDD. Secondly, for the text vector in RDD, use the map function based at the skilled version to calculate the possibility of textual content samples for every magnificence. Then take the elegance with the maximum chance because the magnificence marks. Finally, the classification consequences for the take a look at samples are saved in HDFS.

Algorithm 2: Parallelization of NB classifier's predicting process based on Spark

Input: the preprocessed test set Begin

Step 1. Present the test text in the matrix form as dataMatrix.

Step 2. Calculate the probability belonging to each class, i.e.,
pi.add(theta.mmul(dataMatrix))

Step 3. Take out the maximum value, i.e., result.argmax()

Output: the classification results

3. PROPOSED APPROACH

The proposed gadget is parallel J48 pruned decision tree Classifier for type of multiple samples. In this method beginning with the statistics pre-processing and characteristic extraction, then making use of the massive records classifiers: Naive bayes and Decision Tree one at a time below Spark surroundings. Finally, the effects are evaluated the use of the accuracy metric.

List of Phases

Dataset
Dataset processing
Parallel j48 pruned decision tree Classifier

3.1. Dataset

The statistics set used for experiments is the Big Data in NAT evaluation polarity dataset. It is constructed by way of community opinions dataset that includes evaluations from text, which spans twenty years, which include around 35 million opinions up to December 2019. The Dataset incorporates the subsequent fields like source port, destination port, NAT supply port, NAT designation port, action, bytes, bytes despatched. Bytes received, packets, Elapsed Time, packets despatched and packets acquired.

3.2. Data Pre-processing

The pre-processing is applied to the dataset earlier than passing to the classifier as a way to cleanse and prepare it for type, it includes:

Removing null evaluations: This consists of disposing of the review that contains a null cost, and after counting the range of critiques that contain empty values.

Tokenization: In this segment, the textual content is split into more than one tokens based totally on separator characters which include white space, comma, tab, and so on. The following is an instance of tokenization:

Before Tokenization: 'this paper proposes and implements a parallel J48 pruned decision tree algorithm for multi text classification based on Spark, a parallel memory computing platform for big data'.

After:' this',' paper',' proposes',' and ',implements',' a',' parallel',' naive', 'Bayes',' algorithm', 'for',' Chinese', 'text',' classification',' based',on ',Spark',' a',' parallel',' memory',' computing',' platform',' for',' big',' data'.

Noise removal: This step encompasses cleaning the text from some beside the point facts which may additionally decrease the performance of the classifier which includes numbers, punctuation marks, URL hyperlinks, and special characters.

3.4. Parallel j48 pruned decision tree Classifier

The proposed parallel j48 pruned decision tree set of rules method makes use of multiple distance capabilities, each of which is described on one heterogeneous view of the data. One distance feature on affected

person cases can be described on text; one distance function may be defined on patients' genetic chance profiles; any other distance feature may be described on trajectories of sure biomarker; and so forth. In this case, it's far obviously difficult to define one single distance characteristic primarily based on all of those heterogeneous perspectives. However, one-of-a-kind distance features can be described on extraordinary views of the given facts, such that one distance function represents the view upon which the characteristic is described. Therefore, an important thing of this proposed parallel j48 pruned decision tree method is to learn the load of each distance characteristic this is defined on each view. Furthermore, the weights of distance functions ought to no longer stay unchanged for one-of-a-kind unknown text instances.

The parallel j48 pruned decision tree approach may be described inside the following way. Given an unknown example, the method first learns the burden of every distance function that parallel j48 pruned decision tree: An Enhanced parallel j48 pruned decision tree Approach is described on each view of the statistics. The weight of a distance characteristic is determined by way of the labelled representatives of the unknown example with respect to this distance feature. More in particular, the parallel j48 pruned selection tree of the unknown instance, that are determined the use of this distance feature, serve as the labelled representatives of the unknown instance corresponding to this distance feature. There techniques in pruning prop through parallel j48 fundamental are understood as sub tree replacement, it paintings by means of changing nodes in choice tree alongside leaf. The key point in assembly of decision tree is the choice of the quality characteristic to rip the believed node.

Proposed Algorithm:

Input: parallel J48 Decision Tree T
Procedure PostPruning(Data, TreeSplits)
SplitData(TreeSplits, Data, GrowingSet, PruningSet)
Estimate = DivideAndConquer(GrowingSet)
loop
NewEstimate = Selection(Estimate,PruningSet)
if Accuracy(NewEstimate, PruningSet) < Accuracy(Estimate,PruningSet)
exit loop Estimate = NewEstimate
return(Estimate)
Procedure DivideAndConquer(Data)
Estimate = Ø
while Positive(Data) != Ø
Leaves = Ø
Instance = Data
while Negative(Instance) != Ø
Leaves = Ø
Instance = Data
Leaves = Leaves ∪Find (Leaves, Instance)
Instance = Instance (Leaves,Instance)
Estimate = Estimate ∪ Leaves Data = Data - Instance return (Estimate)
Output: REP TREE

4. EXPERIMENTAL RESULTS AND DISCUSSIONS

The most usually metric used to determine the performance of classifier is accuracy. Since the accuracy is beside the point when records are imbalanced and used some other metrics to evaluate the overall performance. The widespread method for evaluating classifier on imbalanced elegance is Receiver Operating Characteristic.

Here examine proposed and current algorithms are 1. Naive Bayes 2. Parallel j48 pruned decision tree

Naive Bayes:

```
Time taken to build model: 0.22 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0.35 seconds

=== Summary ===

Correctly Classified Instances       12077               92.1486 %
Incorrectly Classified Instances      1029                7.8514 %
Kappa statistic                          0.8737
Mean absolute error                      0.0393
Root mean squared error                  0.1967
Relative absolute error                 13.5846 %
Root relative squared error             51.7439 %
Total Number of Instances            13106

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0.867     0.000     1.000     0.867     0.929      0.857     0.999     1.000     allow
               1.000     0.001     0.996     1.000     0.998      0.997     0.999     0.993     drop
               0.994     0.005     0.985     0.994     0.989      0.986     0.999     0.997     deny
               0.200     0.074     0.002     0.200     0.004      0.013     0.884     0.004     reset-both
Weighted Avg.  0.921     0.001     0.995     0.921     0.955      0.913     0.999     0.997

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
 6547    0   38  966 |    a = allow
    0 2569    0    0 |    b = drop
    0   11 2959    6 |    c = deny
    0    0    8    2 |    d = reset-both
```

Parallel j48 pruned decision tree

```
Time taken to build model: 1.15 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0.04 seconds
=== Summary ===

Correctly Classified Instances       13081               99.8092 %
Incorrectly Classified Instances        25                0.1908 %
Kappa statistic                          0.9967
Mean absolute error                      0.0017
Root mean squared error                  0.0307
Relative absolute error                  0.5732 %
Root relative squared error              8.0668 %
Total Number of Instances            13106

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.001      0.999   1.000      1.000   0.999      1.000     1.000    allow
                 1.000    0.001      0.997   1.000      0.998   0.998      1.000     0.997    drop
                 0.995    0.001      0.996   0.995      0.996   0.995      0.999     0.995    deny
                 0.100    0.000      1.000   0.100      0.182   0.316      0.853     0.103    reset-both
Weighted Avg.    0.998    0.001      0.998   0.998      0.998   0.997      1.000     0.997

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
 7549    0    2    0 |    a = allow
    0 2569    0    0 |    b = drop
    5    9 2962    0 |    c = deny
    0    0    9    1 |    d = reset-both
```

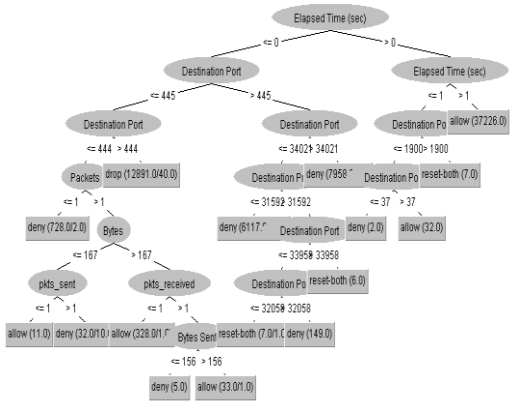

Figure 4.1Parallel j48 pruned decision tree

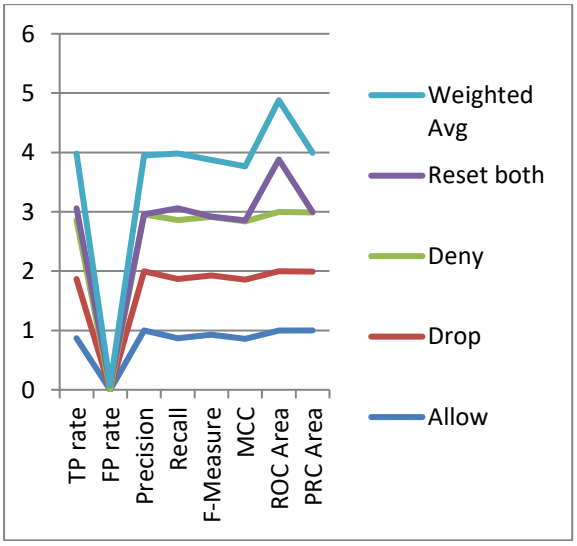

Figure 4.2 Naive bayes

Figure 4.3Parallel j48 pruned decision tree

Accuracy: Proposed algorithm outperforms the others in almost all of the instances. Our proposed linear structure to its bushes instead of the preceding tree shape with a view to reduce gets entry to instances to look nodes. As a result, its benefits have a positive impact on decreasing runtime in complete experiments. Especially because the minimal aid threshold becomes lower, the difference of runtime among our algorithm and the others is greater.

Precision: Proposed set of rules shows the first-rate Precision at the same time as the others have extraordinarily bad performance, which suggests that our scheme can save these increasing attributes greater effectively than the opposite systems of the competitor algorithms. Through the above experimental results, we recognise that the proposed algorithm outperforms the others with recognize to growing transactions and items in terms of scalability in addition to runtime and reminiscence utilization for the real datasets.

Recall: Through the above experimental consequences understand that the proposed algorithm outperforms the others with admire to growing transactions and items in terms of scalability as well as runtime and memory usage for the real datasets.

CONCLUSION

The fast improvement of data mining technology has brought a large quantity of textual content records. This paper aims to categorise the textual content in view of the growing variety of multi texts inside the

method of supervision and the growing demand of processing multi texts, and to improve the efficiency of data query and control. The class performance of the text device is tested and evaluated by using the document of regulatory records. Two classifiers have been compared in phrases of accuracy, Naïve Bayes and Parallel j48 pruned Decision Tree. The experiments results confirmed that the help vector system classifier has higher performance than the other classifiers. As destiny work, further experiments might be carried out using exclusive function units that could beautify the overall performance of the classification.

REFERENCE

[1]. Liu, P., Zhao, H., Teng, J. et al. Parallel naive Bayes algorithm for large-scale Chinese text classification based on spark. J. Cent. South Univ. 26, 1–12(2019). https://doi.org/10.1007/s11771-019-3978-x.

[2]. Raj, S., Ramesh, D., Sreenu, M. et al. EAFIM: efficient apriori-based frequent itemset mining algorithm on Spark for big transactional data. Knowl Inf Syst 62, 3565–3583 (2020). https://doi.org/10.1007/s10115-020-01464-1.

[3]. AlJame, M., Ahmad, I. & Alfailakawi, M. Apache Spark Implementation of Whale OptimizationAlgorithm. ClusterComput 23, 2021–2034(2020). https://doi.org/10.1007/s10586-020-03162-7.

[4]. Mostafaeipour, A., Jahangard Rafsanjani, A., Ahmadi, M. et al. Investigating the performance of Hadoop and Spark platforms on machine learning algorithms. J Supercomput (2020). https://doi.org/10.1007/s11227-020-03328-5.

[5]. Hussain, D.M., Surendran, D. The efficient fast-response content-based image retrieval using spark and MapReduce model framework. J Ambient Intell Human Comput (2020). https://doi.org/10.1007/s12652-020-01775-9.

[6]. Ahmad, A., Paul, A., Din, S. et al. Multilevel Data Processing Using Parallel Algorithms for Analyzing Big Data in High-Performance Computing. Int J Parallel Prog 46, 508–527 (2018). https://doi.org/10.1007/s10766-017-0498-x.

[7]. Cores, F., Guirado, F. & Lerida, J.L. High throughput BLAST algorithm using spark and cassandra. J Supercomput (2020). https://doi.org/10.1007/s11227-020-03338-3.

[8]. González Ordiano, J.Á., Bartschat, A., Ludwig, N. et al. Concept and benchmark results for Big Data energy forecasting based on Apache Spark. J Big Data 5, 11 (2018). https://doi.org/10.1186/s40537-018-0119-6.

[9]. Li, H., Wang, H., Fang, S. et al. An energy-aware scheduling algorithm for big data applications in Spark. Cluster Comput 23, 593–609 (2020). https://doi.org/10.1007/s10586-019-02947-9.

[10]. Shen, J., Wang, H.H. Fusion effect of SVM in spark architecture for speech data mining in cluster structure. IntJSpeech Technol 23, 481–488 (2020). https://doi.org/10.1007/s10772-020-09710-1.

[11]. Xia, D., Ning, F. & He, W. Research on Parallel Adaptive Canopy-K-Means Clustering Algorithm for Big Data Mining Based on Cloud Platform. J Grid Computing 18, 263–273 (2020). https://doi.org/10.1007/s10723-019-09504-z.

[12]. Yuan, J. An Anomaly Data Mining Method for Mass Sensor Networks Using Improved PSO Algorithm Based on Spark Parallel Framework. J Grid Computing 18, 251–261 (2020). https://doi.org/10.1007/s10723-020-09505-3.