Scalable Decision Tree Induction For Mining Big Data



Shabnam Sabah (ID: 011141002)
Sara Zumerrah Binte Anwar (ID: 011133019)
Sadia Afroze (ID: 011133005)
Snigdha Sarker (ID: 011141057)
Department of Computer Science and Engineering
United International University

A thesis submitted for the degree of
BSc in Computer Science & Engineering
May 2019

Abstract

Big data mining is one of the major challenging research issues in the field of machine learning for data mining applications in this present digital era. Big data consists of 3V's: (1) volume - massive amount of data/ too many bytes, (2) velocity - high speed streaming data/ too high a rate, and (3) variety - data are coming from different sources/ too many sources. Collecting and managing real-life big data is a difficult task, as big data is so big that we cannot keep all the data together in a single machine. Therefore, we need advanced relational database management systems with parallel computing to deal with big data. Knowledge mining from big data employing traditional machine learning and data mining techniques is a big issue and attract computational intelligent researcher in this area. In this paper, we have used the decision tree (DT) induction method for mining big data. Decision tree induction is one of the most preferable and well-known supervised learning technique, which is a top-down recursive divide and conquer algorithm and require little prior knowledge for constructing a classifier. The traditional DT algorithms like Iterative Dichotomiser 3 (ID3), C4.5 (a successor of ID3 algorithm), Classification and Regression Trees (CART) are generally built for mining relatively small datasets. So, we need a more scalable decision tree learning approach for mining big data. In this paper, we have engendered several trees employing two scalable decision tree algorithms: RainForest Tree and Bootstrapped Optimistic Algorithm for Tree construction (BOAT) using seven benchmark datasets from Keel Repository and UCI Machine Learning repository. We have compared the performance of RainForest and BOAT algorithms. Also, we have proposed a decision tree merging approach, as decision tree merging is a very complex and challenging task.

Acknowledgements

First and foremost we are extremely thankful to Almighty Allah. We would like to thank our thesis supervisor Dr. Dewan Md. Farid, Associate Professor of Computer Science and Engineering Department, United International University, for his valuable time, effort, and patience, and for giving us an opportunity to do our Bachelor's thesis.

We would like to thank our academic advisor Dr. Swakkhar Shatabda, Associate Professor and Undergraduate Program Co-ordinator of Computer Science and Engineering Department, United International University, for his time and advice throughout our studies.

We would like to thank our parents, family and friends for their constant encouragements and supports throughout my studies at United International University. This work could not have been accomplished without you.

Contents

List of Figures vi

List of Tables vii

1 Introduction 1
- 1.1 Motivation . . . 2
  - 1.1.1 Big Data . . . 2
  - 1.1.2 Data Mining . . . 3
- 1.2 Objectives of the Thesis . . . 3
- 1.3 Thesis Contributions . . . 3
- 1.4 Organization of the Thesis . . . 4

2 Related Work 5
- 2.1 Big Data . . . 5
- 2.2 Decision Tree . . . 6
- 2.3 Summary . . . 6

3 Classifying Big Data Using Decision Trees 7
- 3.1 RainForest . . . 7
- 3.2 Bootstrapped Optimistic Algorithm For Tree Construction (BOAT) . . . 7
- 3.3 Proposed Tree Merging Method . . . 8
  - 3.3.1 Conversion of Decision tree into Decision table . . . 9
  - 3.3.2 Intersection . . . 9
  - 3.3.3 Filtering . . . 10
  - 3.3.4 Reduction . . . 10
- 3.4 Summary . . . 14

4 Experimental Analysis 15
- 4.1 Experimental Setup . . . 15
- 4.2 Datasets . . . 15
- 4.3 Experimental Results . . . 15
- 4.4 Summary . . . 17

5 Conclusions, Limitations and Future Work **18**
5.1 Conclusions 18
5.2 Limitations 18
5.3 Future Work 18

Bibliography **19**

List of Figures

3.1 BOAT - Bootsrapped Optimistic Algorithm for Tree Construction. . . . 8
3.2 Proposed tree merging process. 8
3.3 Decision Tree From Sub-dataset 1. 9

4.1 Performance of RainForest vs BOAT. 17

List of Tables

3.1 Decision table of Subtree 1. 10
3.2 Decision table of Subtree 2. 11
3.3 Intersection of Two Decision Tables. 11
3.4 Filtering 11
3.5 Reduction. 13

4.1 Dataset Description 15
4.2 Results Of RainForest. 16
4.3 BOAT_result. 16

List of Algorithms

1 Insertion Of Two Decision Tables. 12
2 Filtering Of A Table 12
3 Reduction Of A Table 14

Chapter 1

Introduction

Information Technology (IT) has developed so rapidly that resulted in engendering big data over the last decades. It has brought many new challenges as a result of evolution in web technologies, social media, multimedia, mobile and sensing devices etc. For example, Facebook processes 2.5 billion pieces of content and 500+ terabytes of data each day, and Twitter processes 100 million tweets per day that's roughly 13 Gigabytes a day. These big data have accepted an enormous amount of attention by both the industrial and academic fields. Big data is a huge amount of structured and unstructured data collected from a variety of sources. Generally, big data is defined by 3V's: Volume, Variety, and Velocity that describe the amount of data, the type of data, and the speed of incoming and outgoing data. We can add a few more V's like: Vision (having an objective), Verification (making sure that the data comply with a set of specifications) and Validation (examining that its objective is fulfilled). Mining big data using machine learning (ML) and data mining (DM) techniques involve constructions and study of machine intelligent systems that learn from a large amount of data. In big data analytics, data mining is used to extract knowledge, to identify patterns, rules or associations and to expose massive hidden information from huge volumes of complicated data; alternatively machine learning is used to look at patterns and learn from them to adapt behaviour for future events.

Decision tree (DT) induction is the most commonly used supervised learning technique that already used in many real-world applications. DT can discover useful patterns and information from the large and complex datasets. DT can handle numerical, nominal, and textual data with higher accuracy. The goal of a decision tree is to accurately classify the class-labels of new/ unknown instances whose input attribute values are known, but class-labels are unknown. DT is a form of data analysis that extracts models (called classifier) from a given set of training instances. ID3, C4.5 and CART are the traditional decision tree induction algorithms. Each of these approaches uses a feature selection technique to find the root node and internal nodes of the tree. ID3, C4.5, and CART use Information Gain, Gain-Ratio, and Gini Index respectively as feature selection approach to select the best feature. A decision tree algorithm itera-

tively split the given training dataset into smaller sub-datasets until all the instances in a subset refer to only one class or the terminating standard of tree met. A decision tree can deal with noisy datasets and irregular interconnections between input features do not impact the tree performance.

In our thesis, we have analysed and compared the performance of RainForest Tree and Bootstrapped Optimistic Algorithm for Tree construction (BOAT) methods. RainForest and BOAT address the problem of mining big data applying decision trees. Existing decision tree algorithms like ID3, C4.5, and CART become very inefficient for mining big data due to the interchange of the training instances in and out of computer memory. RainForest uses *AVC-table* (Attribute-Value, Class-label) table for each input feature in training data at each decision tree node. The AVC-set counts class labels for each feature-value of each feature for each node of the tree. Basically, RainForest tree updates the AVC-table for each feature of big data and then apply any tree construction approach (like ID3, C4.5, and CART) for building the decision tree. On the contrary, BOAT creates several small sub-datasets from big data that each of sub-dataset fits in the computer memory and generates a tree from each sub-dataset resulting BOAT creates several trees. Finally, all these decision trees are merged together to construct only one decision tree. As like RainForest, BOAT can also use any feature selection approach (like ID3, C4.5, and CART). We have proposed four steps decision tree merging technique that includes decision table, intersection, filtering and reduction in this paper. We have built several decision trees using RainForest and BOAT algorithms on seven benchmark datasets from Keel Repository and UCI Machine Learning Repository. BOAT is three times faster than RainForset tree and the experimental result proved that BOAT achieved a better result in compare with RainForest tree.

1.1 Motivation

1.1.1 Big Data

Over the last few decades Information technology has developed so rapidly that has resulted in data growth on massive scale, which has brought many new challenges as a result of evolution in web technologies, social media, multimedia, mobile and sensing devices. For example, Twitter processes over 70M tweets per day [1]; thereby generating over 8TB daily. These big data have received great amount of attention by both the academic and the industrial world in recent years. Big data is a massive volume of both structured and unstructured data that is very large to process. Big data can also be defined by three V's : Volume, Variety and Velocity which describes the amount of data, the class of data, and the speed of incoming and outgoing data). We can add a few more V's which can also represent the Big Data like, Vision (having an objective), Verification (making sure that the data comply with a set of specifications) and Validation (examining that its objective is fulfilled). The concept of Big Data is defined by Gantz and Reinsel as *Big Data Technologies are new generation technologies and architectures which were designed to extract value from multivariate high volume*

data sets efficiently by providing high speed capturing, discovering and analyzing.

1.1.2 Data Mining

Data mining is the most widely known method in machine learning which is used to extract knowledge, to identify patterns, rules or associations and to uncover large hidden information among huge volumes of complex data. In science and engineering mining big data has so many techniques, among them decision tree is the most widely known approach. Decision trees can discover useful patterns and information from the large and complex set of data. There are many advantages of decision trees in data mining, it is easier to understand which can handle both numerical, nominal and textual data input, able to process erroneous datasets, has higher performance with lesser efforts.

1.2 Objectives of the Thesis

Data mining methods are used for analyzing patterns, rules or associations among huge volume of data across many directions. Nowadays, data mining methods with explanation capability are highly used across many regulations and decision tree induction is one of the methods that have explanation capability. ID3, C4.5 and CART are the traditional decision tree algorithms which are well known for comparatively small datasets. When these algorithms are applied to the mining of very large scale real life database then decision tree induction becomes very inefficient because of swapping training instances in main memory. To be fit into the main memory we need more scalable approaches in order to handle large amount of data. In order to meet this requirement a scalable decision tree induction algorithm which is known as bootstrapped optimistic algorithm for tree construction (BOAT) has already implemented where several trees are scanned to build a new tree that is very close to the tree that would have been generated if all the training data had stored in memory. But instead of examining trees we've implemented an approach of merging trees, as the merging of decision trees is a topic lacking a general data mining approach.

1.3 Thesis Contributions

We've implemented BOAT with a four steps tree merging process which are, converting a decision tree into decision table, intersection, filtering and reduction.

1.4 Organization of the Thesis

The thesis is organized as follows:

Chapter 2 provides related works.

Chapter 3 presents classifying Big Data Using Decision Trees.

Chapter 4 discusses the results and experimental analysis.

Chapter 5 presents the conclusions, summaries the thesis contributions, and discusses the future works.

Chapter 2

Related Work

In recent years, big data are increasing in high speed and becoming an interesting research area. The importance of big data does not turn around how many or how much data we have, but what we have done with it. We can collect data from any repository and analyse for any purpose. Mining data with decision tree is the most popular machine learning algorithm. RainForest and BOAT are scalable decision tree approach that use traditional decision tree induction algorithm for extracting knowledge from big data. In this section, we have presented the related works on Big data, RainForest and BOAT algorithms.

2.1 Big Data

Ramírez-Gallego et al. [2] enumerated and analysed standard machine learning (ML) libraries in MapReduce (MR) programming model for big data analytics. The objective is to provide an introduction of the characteristics of ML algorithms for designing novel algorithms for mining big data. Venturini et al. [3] presented an ensemble learning model named Distributed Associative Classifier (DAC) to allocate the training of an associative classifier among corresponding workers and enhance the final quality of the model. This proposed method embraces various novel techniques to reach high scalability without sacrificing the quality of classification rules. López et al. [4] proposed a new cost sensitive fuzzy rule based classification system for classifying imbalanced big data called Chi-FRBCS-BigDataCS employing MapReduce framework. Wu et al. [5] proposed HACE theorem that specified the characteristics of the Big Data revolution, and introduced a three tire structured big data processing model, from the data mining aspects. Yang and Fong [6] proposed a new incremental decision tree learning algorithm that constructs decision tree from a large set of dataset and compared it with other existing decision tree algorithms under imperfect data streams environment. Rahman et al. [7] proposed an ANN based electricity generation forecasting system to improve the efficiency power management system.

Hashem et al. [8] proposed a classification model for big data and compared it with

several characteristic big data cloud platforms. Fan et al. [9] mainly discussed the challenges and approaches of big data and its current status, controversy and forecast to the future. L'heureux et al. [1] presented emerging machine learning techniques and discussed the several challenges and how to handle those challenges in Big Data. The main purpose is to provide a strong foundation for data mining and machine learning for the future research field. Özköse et al. [10] presented a survey on big data research by introducing prominence's of big data, tools and techniques of mining big data. Petra Perner [11] presented a new quantitative measure technique that explained transformations in the construction of a tree for helping the expert to clarify the dissimilarities of two learnt trees from the same domain. Raghupati et al. [12] explored the emergent fields of big data in healthcare by analysing the advantages, and defined the architectural framework and methodology.

2.2 Decision Tree

Gehrke et al. [13] proposed the fastest scalable classifier that provides notable improvement of performance than the Sprint classification algorithm. Jin et al. [14] proposed a new approach to construct a decision tree which focuses on particular parallel efficiency. The main concept of their approach is to combine RainForest based AVC groups with Sampling, for achieving memory efficient processing of numerical attributes. Gehrke et al. [15] are the first ones who introduced the new scalable algorithm named BOAT that constructs several levels of decision trees in only two scans over the training database. This algorithm can handle a huge range of splitting criteria and faster than the best subsisting algorithms while generating exactly the same decision tree. Yingchun et al. [16] proposed an improvised BOAT classification algorithm that decreases the amount of finding best splitting point calculation and compared it with the original BOAT algorithm and other typical classification algorithms. Dalibard et al. [17] proposed BOAT algorithm to build a Bespoke Auto-tuners that overcome the optimization failure of Black-Box. Authors also proposed Structured Bayesian Optimization (SBO) by using probabilistic programming library. Sherly et al. [18] applied the BOAT algorithm on Credit card dataset to construct an effective fraud detection system that is the combination of clustering and classification technique.

2.3 Summary

In this chapter, mainly we mention some related work of these two well known algorithms for big data mining with decision tree, BOAT and RainForest algorithms.

Chapter 3

Classifying Big Data Using Decision Trees

3.1 RainForest

When a dataset is so large and has a small memory, all of the information or entire dataset does not fit in the memory. To solve this problem, at first Gherke et al. [13] proposed a new approach to manage the scalability of decision trees algorithms. It's another approach to handle the construction of DTs with big data. RainForest adjusts the size of the available main memory and employs to any DT algorithms. For constructing DT algorithms, RainForest is the unifying framework which separates scalability to determine the quality of the tree. The entire dataset is not required in this method. The AVC-set of an attribute, A at node N provides the class label counts for each value of A for the instances at N. AVC-set can easily fit in the memory.

3.2 Bootstrapped Optimistic Algorithm For Tree Construction (BOAT)

Unlike other decision tree algorithm BOAT approaches completely different extension-type procedure. It doesn't use the specific data structure, moreover it uses one statistical approach which is bootstrap. In previous decision tree algorithms, they use the technique of scanning over the database once for each level of the tree but BOAT makes one scan over the dataset for constructing several levels of a tree. BOAT can reduce the data scan and storage requirements. The advantage of BOAT is for the training data, it can take new insertion and update the decision tree without having to reconstruct the whole tree from the very beginning. BOAT algorithm splits the original data into several smaller subsets each of which is adjustable in memory. A tree is generated from each subset, resulting in several decision trees. The trees are examined and used to generate a new tree that would be very close to the tree if all the original training

data had stored in memory. The basic mechanism of BOAT algorithm is shown in Figure 3.1.

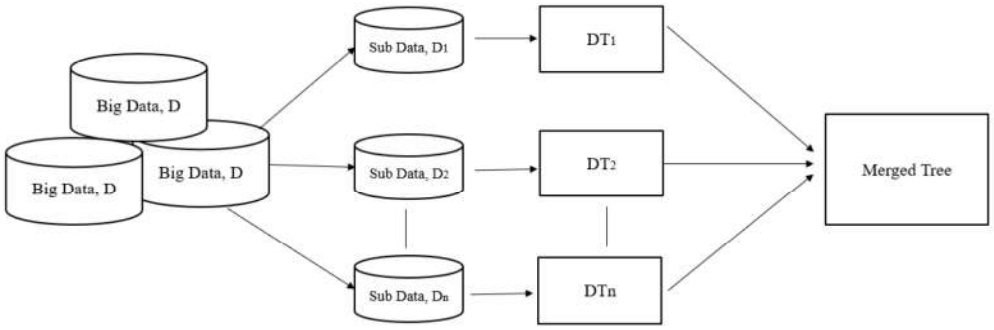


Figure 3.1: BOAT - Bootsrapped Optimistic Algorithm for Tree Construction.

- Suppose the original dataset is D. Now we will divide the dataset into several sub datasets, $D_1, D_2, ..., D_n$.
- Now from each sub dataset $D_1, D_2, ..., D_n$ we will build decision trees $T_1, T_2, ..., T_n$ respectively using C5.0 algorithm.
- Then we will merge the decision tress $T_1, T_2, ..., T_n$ and create a new tree T' which will resemble the tree that would have been generated from the original data set D.

3.3 Proposed Tree Merging Method

In this section, we have implemented the BOAT with a four step tree merging process: (1) Conversion of decision tree into decision table, (2) Intersection, (3) Filtering, and (4) Reduction. Figure 3.2 shows the process of tree merging.

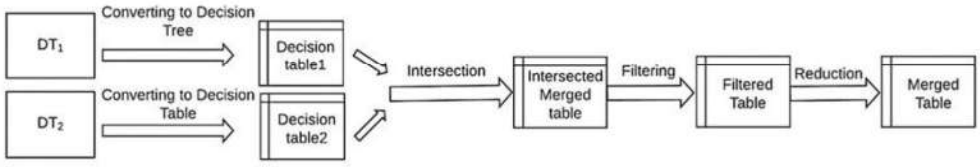


Figure 3.2: Proposed tree merging process.

3.3.1 Conversion of Decision tree into Decision table

A decision table is another way to represent a DT into a different data-structure. In the decision table, a region or a row defines the path of a tree from the root node of leaves. Columns identify the class, weight and set of values of each attribute. Figure 3.3 shows the DT of a sub-dataset of 50 instances of famous Iris dataset and Table 3.1 is the corresponding decision table. Similarly, Table 3.2 is the decision table for other different sub-dataset of the same data that belongs to 33 instances. We can construct more decision table in a same way. In the context of a decision table weight is a comparative measurement of the importance of a decision region which can be defined as the ratio of example in the region compared to the whole dataset [19] [20].

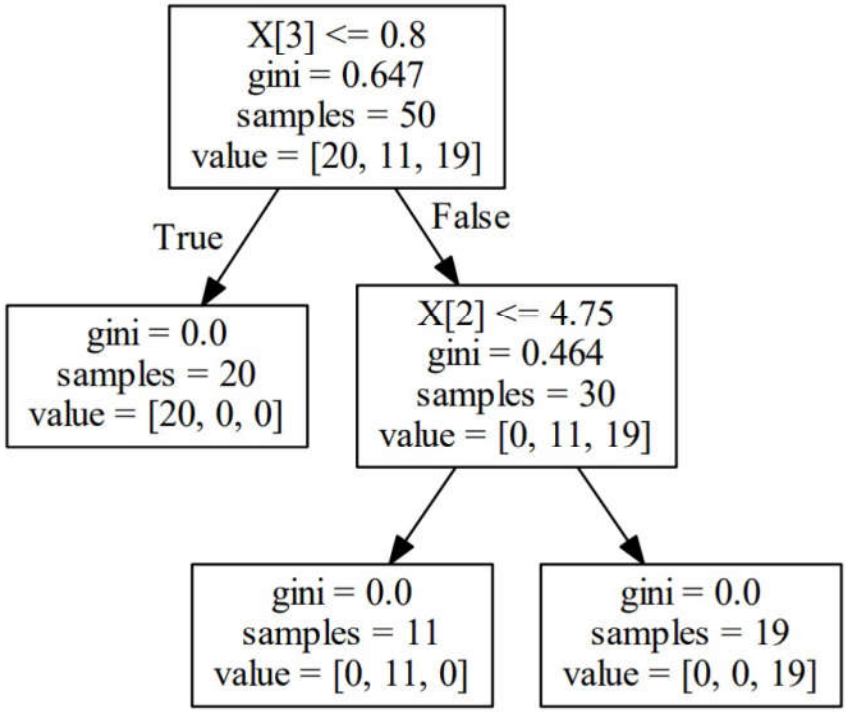


Figure 3.3: Decision Tree From Sub-dataset 1.

3.3.2 Intersection

In this step of the process, we enumerate the cross product of two decision tables in order to merge them into one table using our intersection method in Algorithm 1 and this process is known as the

intersection. The resulted table contains attributes of both tables. For each pair of regions, the set of values of each attribute is intersected and may have the following outcomes:

- If there are common values for the attribute being intersected in both regions, then these are put into the attribute in the merged region.
- If there are no common values then they are considered as disjoint regions.
- The set of values in the other table is copied to the merged region if the attribute does not have a value or is not present in one of the table.

Each region in the intersected table must have a weight for resembling the same format of the decision table. For evaluation purposes, we calculate the weight by taking the average of two decision tables. The approach of assigning a class to the merged region is forthright. If the class value is same in both regions, we assign the class directly to the region. If the class conflicts, the class with the maximum weight between the two regions is assigned. Table 3.3 depicts the intersection of Tables 3.1 and 3.2 [19] [20].

3.3.3 Filtering

Filtering is another sub-process of merging. We remove the disjoint regions from the intersected table as they refer to pairs of regions of the main tables that have no common values. Then the weight of each region is normalized to gain a total weight of 100% for the table. Table 3.4 shows the filtered table of Table 3.3 that we built by our proposed filtering criteria in Algorithm 2. However, if all regions of the intersected table are disjoint regions then we cannot go further with this merging process as the filtered table will be empty, and in that case, merging process will halt [19] [20].

3.3.4 Reduction

To obtain a simpler form of the final merged table by the reduction we bound the number of regions in the filtered table. In this process, we scan to find out which regions can be merged together. Here we consider for numerical variables, if the regions have the same class and we can group their intervals into one interval then, we join those regions into one region and assign the class into that region. If the regions have the same class but different intervals, then we assign that class into separate regions. After that, we update the weight of the final merged table by summing the weight of the joined regions. The regions that have been chosen to reduction are again scanned and those that have zero weight are eliminated. In reduction, there can be some attributes with the equivalent value in all decision regions, the columns for these attributes are eliminated from the table. Hence, all those ideas are adopted into

Table 3.1: Decision table of Subtree 1.

Region	Class	Weight	Variables	
			X[2]	X[3]
1	0	40	[0,0]	[0,0.8]
2	1	22	[0,4.75]	[0.8,∞]
3	2	38	[4.75,∞]	[0.8,∞]

Table 3.2: Decision table of Subtree 2.

Region	Class	Weight	Variables		
			X[0]	X[2]	X[3]
1	0	21	[0,0]	[0,2.45]	[0,1.6500001]
2	1	42	[0,0]	[2.45,∞]	[0,1.6500001]
3	2	3	[0,5.95]	[0,0]	[1.6500001,∞]
4	1	3	[0,5.95]	[0,0]	[1.75,∞]
5	2	30	[5.95,∞]	[0,0]	[1.6500001,∞]

Table 3.3: Intersection of Two Decision Tables.

Region	Class	Weight	Variables		
			X[0]	X[2]	X[3]
1	0	30	[0,0]	[0,2.45]	[0,0.8]
2	1	41	[0,0]	[2.45,∞]	[0,0.8]
3	0	21	[0,5.95]	[0,0]	ϕ
4	0	21	[0,5.95]	[0,0]	ϕ
5	0	35	[5.95,∞]	[0,0]	ϕ
6	1	21	[0,0]	[0,2.45]	[0.8,1.6500001]
7	1	32	[0,0]	[2.45,4.75]	[0.8,1.6500001]
8	1	12	[0,5.95]	[0,4.75]	[1.6500001,1.75]
9	1	12	[0,5.95]	[0,4.75]	[1.75,∞]
10	2	26	[5.95,∞]	[0,4.75]	[1.6500001,∞]
11	2	29	[0,0]	ϕ	[0.8,1.6500001]
12	1	40	[0,0]	[4.75,∞]	[0.8,1.6500001]
13	2	20	[0,5.95]	[4.75,∞]	[1.6500001,1.75]
14	2	20	[0,5.95]	[4.75,∞]	[1.75,∞]
15	2	34	[5.95,∞]	[4.75,∞]	[1.6500001,∞]

Table 3.4: Filtered Table

Region	Class	Weight	Variables		
			X[0]	X[2]	X[3]
1	0	11	[0,0]	[0,2.45]	[0,0.8]
2	1	15	[0,0]	[2.45,∞]	[0,0.8]
3	1	8	[0,0]	[0,2.45]	[0.8,1.6500001]
4	1	12	[0,0]	[2.45,4.75]	[0.8,1.6500001]
5	1	5	[0,5.95]	[0,4.75]	[1.6500001,1.75]
6	1	5	[0,5.95]	[0,4.75]	[1.75,∞]
7	2	10	[5.95,∞]	[0,4.75]	[1.6500001,∞]
8	1	14	[0,0]	[4.75,∞]	[0.8,1.6500001]
9	2	7	[0,5.95]	[4.75,∞]	[1.6500001,1.75]
10	2	7	[0,5.95]	[4.75,∞]	[1.75,∞]
11	2	12	[5.95,∞]	[4.75,∞]	[[1.6500001,∞]

Algorithm 1 Insertion Of Two Decision Tables.

```
1: Input: t1 = table1, t2 = table2
2: Output: t = table (Intersecting table of table1 and table2)
3: Method:
4: M ← length(t1);
5: N ← length(t1);
6: row ← 1,
7: for i ← 1, M do
8:     for j ← 1, N do
9:         t[row][weight] ← average(t1[i][weight], t2[j][weight]) ;
10:        t[row][class] ← maxWeight(t1[i][class], t2[j][class]);
11:        t[row] ← union.featureValue(t1[i], t2[j]);
12:        row ← row + 1
13:    end for
14: end for
15: return t;
```

Algorithm 2 Filtering Of A Table

```
1: Input: t1 = table1
2: Output: t = table (filtered Version of table1)
3: Method:
4: M ← length(t1);
5: row ← 1,
6: for i ← 1, M do
7:     if t1[i] is a valid row then
8:         t[row] ← t1[i];
9:         t[row][weight] ← scalingWeight(t1[i][weight])
10:        row ← row + 1;
11:    end if
12: end for
13: return t;
```

the reduction Algorithm 3 and using this we get Table 3.5 which is the final merged table of the filtered Table 3.4.

Table 3.5: Reduction (The Final Merged Table)

Region	Class	Weight	Variables		
			X[0]	X[2]	X[3]
1	0	11	[0,0]	[0,2.45]	[0,0.8]
2	1	59	[0,5.95]	[0,∞]	[0,∞]
3	2	36	[0,∞]	[0,∞]	[1.6500001,∞]

Algorithm 3 Reduction Of A Table

```
1: Input: t1 = table1
2: Output: t = table (reduced version of table1 )
3: Method:
4: M ← length(t1);
5: t ← t1;
6: for i ← 1, M − 1 do
7:     for j ← i + 1, M do
8:         if t1[i] [class]!= -1 and t1[i][class] = t1[j][class] then
9:             if union of t1[i] and t1[j] is possible then
10:                t1[i] ← union(t1[i], t1[j]);
11:                t1[j][class] = -1;
12:                t[i] ← t1[i]);
13:            end if
14:        end if
15:    end for
16: end for
17: for i ← 1, M do
18:    if If t1[i][class] = -1  then
19:        t ← deleteRow(t, t[i]);
20:    end if
21: end for
22: return t;
```

3.4 Summary

This chapter is focused on the two most talked about algorithms BOAT and RainForest which are known for handling decision trees with Big Data. We mainly discussed our approach of implementing BOAT elaborately.

Chapter 4

Experimental Analysis

4.1 Experimental Setup

We have collected seven datasets from Keel and UCI Machine learning repository, the dataset descriptions are shown in Table 4.1. We have applied data preprocessing with dealing missing values, removed duplicate instances, and feature selection. Then, we have implemented the proposed method using Python programming language.

4.2 Datasets

Table 4.1: Data set descriptions.

No	Data Sets	Instances	No of Attribute	Attribute Type	Classes
1	Australian	690	14	Numeric	2
2	Mammographic	961	5	Numeric	2
3	Diabetis	768	8	Numeric	2
4	Breast Cancer	569	30	Numeric	2
5	Thyroid	7200	21	Numeric	3
6	Spambase	497	57	Numeric	2
7	Monk-2	432	6	Numeric	2

4.3 Experimental Results

We have considered accuracy, precision, recall and f-measure that shown in Eq. 4.1 to 4.4 where TP refers to true positive, TN refers to true negative, FP refers to false positive, and FN refers to false negative. Table 4.2 and 4.3 represent the comparative results respectively for RainForest and BOAT algorithms.

$$accuracy = \frac{\sum_{i=1}^{|X|} assess(x_i)}{|X|}, x_i \in X \tag{4.1}$$

$$precision = \frac{TP}{TP + FP} \tag{4.2}$$

$$recall = \frac{TP}{TP + FN} \tag{4.3}$$

$$f - measure = \frac{2 \times precision \times recall}{precision + recall} \tag{4.4}$$

Table 4.2: Performance of RainForest algorithm.

Algorithm	**Classification** Accuracy(%)	**Precision** (weighted avg.)	**Recall** (weighted avg.)	**F-Measure** (weighted avg.)
Austrailian	82.2222	0.822	0.822	0.821
Mammographic	64.7727	0.813	0.648	0.629
Diabetis	72.619	0.723	0.726	0.724
Breast Cancer	91.9075	0.932	0.919	0.922
Thyroid	99.3333	0.994	0.993	0.994
Spambase	89.1667	0.899	0.892	0.893
Monk-2	100	1.000	1.000	1.000

Table 4.3: Performance of BOAT algorithm.

Dataset	**Classification** Accuracy(%)	**Precision** (weighted avg.)	**Recall** (weighted avg.)	**F-Measure** (weighted avg.)
Austrailian	83.3333	0.834	0.833	0.833
Mammographic	61.3636	0.644	0.614	0.617
Diabetis	72.0238	0.711	0.720	0.709
Breast Cancer	94.2197	0.945	0.942	0.943
Thyroid	94	0.914	0.908	0.909
Spambase	91.1111	0.911	0.911	0.911
Monk-2	100	1.000	1.000	1.000

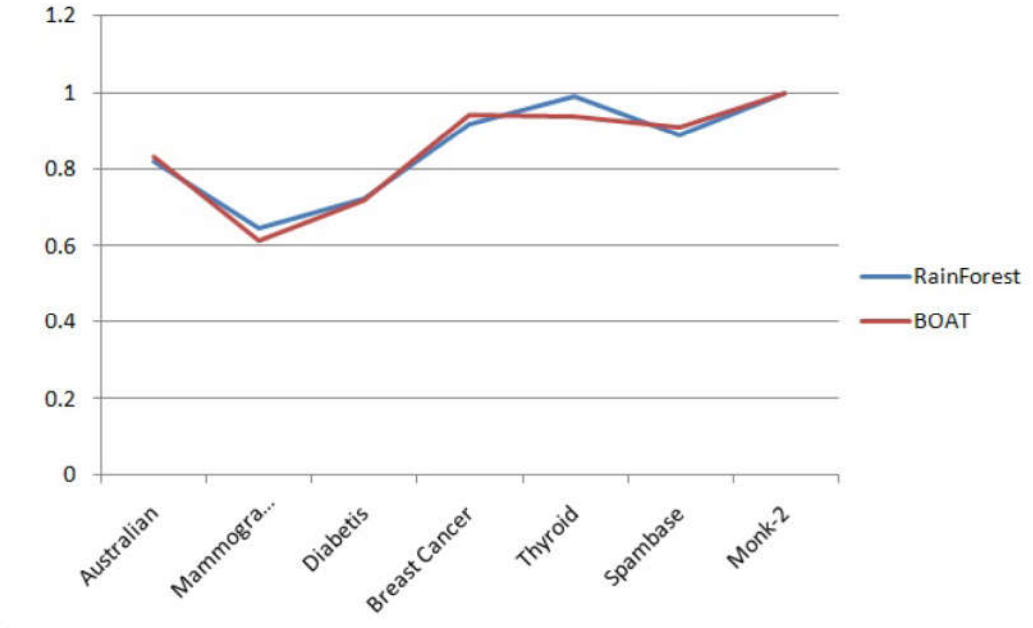


Figure 4.1: Performance of RainForest vs BOAT.

4.4 Summary

In this chapter we've compared the experimental result of BOAT with RainForest algorithm. Here we can see BOAT performing better than RainForest for most of the datasets.

Chapter 5

Conclusions, Limitations and Future Work

5.1 Conclusions

Big data mining is one of the most significant topics that applies machine learning and data mining techniques to extract knowledge from big data. In this paper, we applied decision tree induction for mining big data. We have built and compared the performance of RainForest and BOAT algorithms with benchmark datasets. We have found the BOAT is two/three times faster than RainForest. Also, we have proposed a new tree merging technique in this paper that is very useful for merging several trees together.

5.2 Limitations

In our thesis we have worked with only numeric values in our datasets, but we could have achieved more efficiency in our proposed method if we have also handled the datasets with nominal values. In decision tree algorithms we generally classify the target class from a tree but in our method we classify the target class from the rules of final merged table.

5.3 Future Work

In future, we will work with real-time traffic big data mining employing decision tree induction.

Bibliography

[1] A. L'Heueux, K. Grolinger, H. F. Elymany, and M. A. M. Capretz, "Machine learning with big data: Challenges and approaches," *IEEE Access*, vol. 5, pp. 7776–7797, April 2017. 2, 6

[2] S. Ramrez-Gallego, A. Fernndeza, S. Garcaa, M. Chenb, Francisco, and Herreraa, "Big data - tutorial and guidelines on information and process fusion for analytics algorithms with mapreduce," *Information Fusion*, vol. 42, pp. 51–61, July 2018. 5

[3] L. Venturini, E. Baralis, and P. Garza, "Scaling associative classication for very large datasets," *Journal of Big Data*, vol. 4, pp. 1–24, December 2017. 5

[4] V. Lpez, S. Ro, J. M. Bentez, and FranciscoHerrera, "Cost-sensitive linguistic fuzzy rule based classification systems under the mapreduce framework for imbalanced big data," *Fuzzy Sets and Systems*, vol. 258, pp. 5–38, January 2015. 5

[5] X. Wu, X. Zhu, G.-Q. Wu, and W. Ding, "Data mining with big data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, pp. 97–107, January 2014. 5

[6] H. Yang and S. Fong, "Incrementally optimized decision tree for noisy big data," in *Proceedings of the 1st International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications*. ACM, August 2012, pp. 36–44. 5

[7] M. N. Rahman, A. Esmailpour, and J. Zhao, "Machine learning with big data an efficient electricity generation forecasting system," *Big Data Research*, vol. 5, pp. 9–15, 2016. 5

[8] I. A. T. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani, and S. U. Khan, "The rise of big data on cloud computing: Review and open research issues," *Information Systems*, vol. 47, pp. 98–115, January 2015. 5

[9] W. Fan and A. Bifet, "Mining big data: Current status, and forecast to the future," *ACM SIGKDD Explorations Newsletter*, vol. 14, pp. 1–5, December 2012. 6

[10] H. Özköse, E. S. Arı, and C. Gencer, "Yesterday, today and tomorrow of big data," *Procedia-Social and Behavioral Sciences*, vol. 195, pp. 1042–1050, 2015. 6

[11] P. Perner, "Decision tree induction methods and their application to big data," in *Modeling and Optimization in Science and Technologies, Modeling and Processing for Next-Generation Big-Data Technologies With Applications and Case Studies*, vol. 4. Springer Verlag, January 2015, pp. 57–88. 6

[12] W. Raghupathi and V. Raghupathi, "Big data analytics in healthcare: promise and potential," *Health Information Science and Systems*, vol. 7, pp. 1–10, Febuary 2014. 6

[13] J. Gehrke, R. Ramakrishnan, and V. Ganti, "Rainforest - a framework for fast decision tree construction of large datasets," in *Data Mining and Knowledge Discovery*, vol. 4. SpringerLink, July 2000, pp. 127–162. 6, 7

[14] R. Jin and G. Agrawal, "Communication and memory efficient parallel decision tree construction," in *Proceedings of the 2003 SIAM International Conference on Data Mining*, vol. 4. Siam, May 2003, pp. 1–11. 6

[15] J. Gehrke, V. Ganti, R. Ramakrishnany, and W.-Y. Lohz, "Boatoptimistic decision tree construction," in *SIGMOD '99 Proceedings of the 1999 ACM SIGMOD international conference on Management of data*, vol. 28. ACM Digital Library, June 1999, pp. 169–180. 6

[16] L. Yingchun, C. Meiling, and W. Huiwen, "Improvement and validation of the boat algorithm," *Journal of Multimedia*, vol. 9, pp. 542–547, April 2014. 6

[17] V. Dalibard, M. Schaarschmidt, and E. Yoneki, "Boat: Building auto-tuners with structured bayesian optimization," in *WWW '17 Proceedings of the 26th International Conference on World Wide Web*. ACM Digital Library, April 2017, pp. 479–488. 6

[18] K. K. Sherly and R. Nedunchezhian, "Boat adaptive credit card fraud detection system," *IEEE International Conference on Computational Intelligence and Computing Research*, December 2010. 6

[19] P. Strecht, J. Mendes-Moreira, and C. Soares, "Merging decision trees: A case study in predicting student performance," *International Conference on Advanced Data Mining and Applications*, vol. 8933, pp. 535–548, March 2014. 9, 10

[20] P. Strecht, "A survey of merging decision trees data mining approaches," in *Proceedings of the 10th Doctoral Symposium in Informatics Engineering - DSIE15*, January 2015, pp. 38–47. 9, 10