A DYNAMIC AND SCALABLE EVOLUTIONARY DATA MINING & KDD FOR DISTRIBUTED ENVIRONMENT

Ph.D. Synopsis

Submitted To
Gujarat Technological University

For The Degree
Of
Doctor Of Philosophy
In
Computer/IT Engineering

By
Dineshkumar Bhagwandas Vaghela
Enrollment No: 119997107013
(Computer/IT Engineering)

Supervisor :
Dr. Priyanka Sharma
Professor,
Raksha Shakti University
Gujarat, India

Co-Supervisor :
Dr. Kalpdrum Passi,
Associate Professor,
Mathematics & Computer Science Department,
Laurentian University
Canada
# Index

1  Abstract............................................................................................................................................. 3
2  Brief description on the state of the art of the research topic.................................................. 4
   2.1 Distributed Data Mining........................................................................................................ 4
   2.2 Decision Tree Algorithms.................................................................................................... 5
   2.3 Merging Decision Tree Models.......................................................................................... 7
3  Definition of the problem............................................................................................................ 9
4  Objective and scope of work......................................................................................................... 9
5  Original contribution by the thesis............................................................................................ 10
6  Methodology of research, results / comparisons....................................................................... 10
   6.1 Methodology of research....................................................................................................... 10
   6.2 The Model and its components............................................................................................. 11
      6.2.1 The proposed framework............................................................................................. 11
      6.2.2 The system architecture at local site............................................................................. 14
      6.2.3 The system architecture at coordinator site................................................................. 15
      6.2.4 Decision rule merging policies.................................................................................... 16
   6.3 Results/Comparisons............................................................................................................. 17
7  Achievements with respect to objectives................................................................................... 19
8  Conclusions..................................................................................................................................... 20
9  Copies of papers published and a list of all publications arising from the thesis.......................... 21
10 Patent/Copyright (If any)........................................................................................................... 22
11 Achievements............................................................................................................................ 22
12 Partial List of References........................................................................................................... 22
1 Abstract

The explosive growth in the amount of data in the field of biology, education, environmental research, sensor network, stock market, weather forecasting and many more due to vast use of internet in distributed environment has generated an urgent need for new techniques and tools that can intelligently automatically transform the processed data into useful information and knowledge. Hence data mining has become a research with increasing importance. Since continuation in collection of more data at this scale, formalizing the process of big data analysis will become paramount. Given the vast amount of data are geographically spread across the globe, this means a very large number of models is generated, which raises problems on how to generalize knowledge in order to have a global view of the phenomena across the organization.

Classification and prediction are two forms of data analysis that can be used to extract models describing important data classes or to predict future data trends. Such analysis can help provide us with better understanding of the data at large. Whereas classification predicts categorical (discrete, unordered) labels, prediction models continuous valued functions. Many classification and prediction methods have been proposed by researcher in machine learning pattern recognition and statistics. Most algorithms are memory resident, typically assuming a small data size, static in nature, not scalable and not domain-free. Recent data mining research has built on such work, developing scalable classification and prediction techniques capable of handling large disk-resident data. The application of the classical knowledge discovery process in distributed environments requires the collection of distributed data in a data warehouse for central processing. However, this is usually either ineffective or infeasible for several reasons such as (1) Storage cost (2) Communication cost (3) Computational cost (4) Private and sensitive data. From the literature review, the most promising issues for prediction in distributed environment are RAM size due to very large volume of data set, scalability of size of data set and dynamic nature of learning.

The classification with decision tree learner uses the training data set for learning purpose and generates the decision tree model. The local decision tree model generated at each site is not sufficient to provide the global view for prediction as because the local training data set are not in co-relation with other data set at different geographically spread sites. In order to generate the global decision tree either these all data set need to be collected at one location and then do data mining operation. The other approach is intermediate message passing among the sites involved for training the model. These participating sites have to communicate among each other through passing their intermediate trained models to generate the global model. This leads multiple messages passing which causes the communication overhead. So both the approaches are not effective and efficient, and hence this is the motivation for this research. The objectives of the research are: 1) To minimize the training time and communication overhead and 2) To preserve the prediction quality. We have proposed the effective and efficient approach which works on global decision tree generation in distributed environment to extract the global knowledge.
This research has been carried out on real data set of Parul University for the student’s admission prediction in different fields/branches of different colleges. In the first phase, these data have been collected from the Parul University Web Portal. There are more than 1,00,000 records in total used for training purpose. As data collected from the University Portal they need to pre-process. The data are stored in .csv file format.

In the second phase of this research, at the local site C5.0 algorithm (complexity $O(mn^2)$) generates the decision tree. In third phase the decision tree at each site is converted into decision rules using the proposed parser. These decision rules have been later converted into the decision tables. As the decision tables need to send at the coordinator site, to reduce the transmission cost they have been converted into XML file in the fourth phase. In the fifth phase the global tree model is generated at coordinator site by consolidating the decision tables formed from XML files.

In sixth phase the dataset has been equally partitioned into the subsets equal to the number of sites. The experiments have been performed on 10k, 20k, 50k and 100k records (Here k means thousand) at 2, 5 and 10 sites. The local training models have been generated and merged using the proposed approach. The accuracy of these global models has been checked on test datasets. The accuracy is more than 98% to classify the test dataset. The results of basic comparison clearly show that accuracy, training time, communication overhead and other parameters have been optimized. The data set of student admission for the year 2013-14, 2014-15 have been used to train the model, this model has been used with the data set of student admission for the year 2015-16 which gives more than 98.03% accuracy for the prediction. These experimental results have been also verified using the 10-fold cross validation.

2 Brief description on the state of the art of the research topic

Over the last two decades, researchers have done work on data mining algorithms for analysis and visualization, which are based on centralized, client-server DDM, mobile agent based DDM approaches. Many of the researchers have also worked on distributed classification and prediction with decision tree generation and merging approach. However, no efforts have been put towards the dynamic and scalable approach and moreover to this none of the work is domain-free, preserves the prediction quality and minimizes the training and communication time. As a result during the literature review, we did not get any research paper addressing the above issues in distributed data mining. However, only few inventions have been claimed on this work. Hall Chawla and Bowyer[4], Bursteins and Long [10], Adrzejak, Longer & Zabala [1] and Strecht, Moreira & Soares[2] have claimed decision tree merging approaches. Hence these inventions are not capable of large data size, dynamic, scalable, domain free work which moreover minimizes the training and communication cost and also preserves the prediction quality.

2.1 Distributed Data Mining
The huge datasets have been generated exponentially day to day by the software applications developed for various services such as banking, education, stock market, supermarket and mobile devices. For the analysis and visualization these data need to be processed with
Distributed Data Mining (DDM) approach. Distributed Data Mining can be implemented using one of the four approaches, mentioned below [17].

- **Central approach:** Bring the all site datasets to a central site, and then apply data mining on the entire combined dataset. This causes two problems, first heavy communication cost to bring the entire data to a central site and second data privacy preservation.
- **Merge approach:** Generate the local data model at each site locally. All these models are sent to central site to merge/combine into a global model. This approach is brought into play in the works of [18] [19] [20]. But this approach suffers from scalability problem with increase in number of sites.
- **Sample approach:** The third approach is using samples. A small set of representative data is carefully sampled at each site to form one global representative dataset. Data mining can then be performed on the global representative data set.
- **Intermediate Message Passing Approach:** Unlike all the previous three approaches where a central site to assist the distributed data mining, here in the fourth approach data mining is performed on P2P networks, where sites communicate directly with each other without involving a central server [21][22].

### 2.2 Decision Tree Algorithms

Researchers have developed various decision tree algorithms over a period of time with enhancement in performance and ability to handle various types of data. Some important algorithms are discussed below.

**CHID:** CHAID (CHi-squared Automatic Interaction Detector) is a fundamental decision tree learning algorithm. It was developed by Gordon V Kass [11] in 1980. CHAID is easy to interpret, easy to handle and can be used for classification and detection of interaction between variables. CHID is an extension of the AID (Automatic Interaction Detector) and THAID (Theta Automatic Interaction Detector) procedures. It works on principal of adjusted significance testing. After detection of interaction between variables it selects the best attribute for splitting the node which made a child node as a collection of homogeneous values of the selected attribute. The method can handle missing values. It does not imply any pruning method.

**CART:** Classification and regression tree (CART) proposed by Breiman et al. [12] constructs binary trees which is also refer as Hierarchical Optimal Discriminate Analysis (HODA). CART is a non-parametric decision tree learning technique that produces either classification or regression trees, depending on whether the dependent variable is categorical or numeric, respectively. The word binary implies that a node in a decision tree can only be split into two groups. CART uses gini index as impurity measure for selecting attribute. The attribute with the largest reduction in impurity is used for splitting the node's records. CART accepts data with numerical or categorical values and also handles missing attribute values. It uses cost-complexity pruning and also generate regression trees.

**ID3:** ID3 (Iterative Dichotomiser 3) decision tree algorithm is developed by Quinlan [13]. In the decision tree method, information gain approach is generally used to determine suitable
property for each node of a generated decision tree. Thus, we can select the attribute with the highest information gain (entropy reduction in the level of maximum) as the test attribute of current node. In this way, the information needed to classify the training sample subset obtained from later on partitioning will be the smallest. That is to say, the use of this property to partition the sample set contained in current node will make the mixture degree of different types for all generated sample subsets reduce to a minimum. Therefore, the use of such an information theory approach will effectively reduce the required dividing number of object classification.

**C4.5:** C4.5 is an algorithm used to generate a decision tree developed by Ross Quinlan. C4.5 is an extension of Quinlan’s earlier ID3 algorithm. The decision trees generated by C4.5 can be used for classification, and for this reason C4.5 is often referred to as a statistical classifier [14]. C4.5 algorithm uses information gain as splitting criteria. It can accept data with categorical or numerical values. To handle continuous values it generates threshold and then divides attributes with values above the threshold and values equal to or below the threshold. C4.5 algorithm can easily handle missing values. As missing attribute values are not utilized in gain calculations by C4.5.

**C5.0/Sec 5:** C5.0 algorithm is an extension of C4.5 algorithm which is also extension of ID3. It is the classification algorithm which applies in big data set. It is better than C4.5 on the speed, memory and the efficiency. C5.0 model works by splitting the sample based on the field that provides the maximum information gain. The C5.0 model can split samples on basis of the biggest information gain field. The sample subset that is get from the former split will be split afterward. The process will continue until the sample subset cannot be split and is usually according to another field. Finally, examine the lowest level split, those sample subsets that don’t have remarkable contribution to the model will be rejected. C5.0 is easily handled the multi value attribute and missing attribute from data set [15].

**Hunt's Algorithm:** Hunt’s algorithm generates a Decision tree by top-down or divides and conquers approach. The sample/row data contains more than one class, use an attribute test to split the data into smaller subsets. Hunt’s algorithm maintains optimal split for every stage according to some threshold value as greedy fashion [16]. Table 1 shows the comparisons of different algorithms with different parameters.

<table>
<thead>
<tr>
<th></th>
<th>ID3</th>
<th>C4.5</th>
<th>C5.0</th>
<th>CART</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of data</td>
<td>Categorical</td>
<td>Continuous and Categorical</td>
<td>Continuous and Categorical, dates, times, timestamps</td>
<td>continuous and nominal attributes data</td>
</tr>
<tr>
<td>Speed</td>
<td>Low</td>
<td>Faster than ID3</td>
<td>Highest</td>
<td>Average</td>
</tr>
<tr>
<td>Pruning</td>
<td>No</td>
<td>Pre-pruning</td>
<td>Pre-pruning</td>
<td>Post pruning</td>
</tr>
<tr>
<td>Boosting</td>
<td>Not supported</td>
<td>Not supported</td>
<td>Supported</td>
<td>Supported</td>
</tr>
<tr>
<td>Missing Values</td>
<td>Can’t deal with</td>
<td>Can’t deal with</td>
<td>Can deal with</td>
<td>Can deal with</td>
</tr>
<tr>
<td>Formula</td>
<td>Use information entropy and information Gain</td>
<td>Use split info and gain ratio</td>
<td>Same as C4.5</td>
<td>Use Gini diversity index</td>
</tr>
</tbody>
</table>

Table 1: Comparisons between different Decision Tree Algorithms
2.3 Merging Decision Tree Models

A more common approach is the combination of rules derived from decision trees. The idea is to convert decision trees from two models into decision rules by combining the rules into new rules, reducing their number and finally growing a decision tree of the merged model. The basic fundamentals of the process are first presented in the doctoral thesis of Williams [3] and over the years, other researchers have contributed by proposing different ways of carrying out intermediate tasks.

Provost and Hennessy [8, 9] present an approach to learning and combining rules on disjoint subsets of a full training data. A rule based learning algorithm is used to generate rules on each subset of the training data. The merged model is constructed from satisfactory rules, i.e., rules that are generic enough to be evaluated in the other models. All rules that are considered satisfactory on the full data set are retained as they constitute a superset of the rules generated when learning is done on the full training set. This approach has not been replicated by other researchers. Table 2 summarizes research examples of this approach, specifying the problem (or motivation) and data sets used.

<table>
<thead>
<tr>
<th>Research</th>
<th>Problem/motivation</th>
<th>Data sets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hall, Chawla and Bowyer [4]</td>
<td>Train model in a very large data set</td>
<td>Iris, Pima Indians Diabetes</td>
</tr>
</tbody>
</table>

Table 2. Research examples of combination of rules approaches to merge models

Hall, Chawla and Bowyer [4, 5] research present as rationale that is not possible to train decision trees in very large data sets because it could overwhelm the computer system’s memory by making the learning process very slow. Although a tangible problem in 1998, nowadays, this argument still makes sense as the notion of very large data sets has turned into the big data paradigm. The approach involves breaking down a large data set into n disjoint partitions, then, in parallel, train a decision tree on each. Each model, in this perspective, is considered an independent learner. Globally, models can be viewed as agents learning a little about a domain with the knowledge of each agent to be combined into one knowledge base. Simple experiments to test the feasibility of this approach were done on two datasets: Iris and Pima Indians Diabetes. In both cases, the data sets were split across two processors and then the resulting models merged.

Bursteinas and Long [10] research aims to develop a technique for mining data which is distributed on distant machines, connected by low transparency connections arguing that...
there is a lack of algorithms and systems which could perform data mining under such conditions. The merging procedure is divided into two scenarios: one for disjoined partitions and one for overlapped partitions. To evaluate the quality of the method, several experiments have been performed. The results showed the equivalence of combined classifiers with the classifier induced on a monolithic data set. The main advantage of the proposed method is its ability to induce globally applicable classifiers from distributed data without costly data transportation. It can also be applied to parallelize mining of large-scale monolithic data sets. Experiments are performed merging two models in data sets taken from the UCI Machine Learning Repository [23].

Andrzejak, Langner and Zabala [1] propose a method for learning in parallel or from distributed data. Factors cited as contributing to this trend include emergence of data sets with exceeding RAM sizes and inherently distributed scenarios such as mobile environments. Also in these cases interpretable models are favored: they facilitate identifying artifacts and understanding the impact of individual variables. The method is compared with ensemble learning, because in a distributed environment, even if the individual learner on each site is interpretable, the overall model usually is not, citing as example the case of voting schemes. To overcome the problem they propose an approach for merging of decision trees (each learned independently) into a single decision tree. The method complements the existing parallel decision trees algorithms by providing interpretable intermediate models and tolerating constraints on bandwidth and RAMS size. The latter properties are achieved by trading RAM and communication constraints for accuracy. The method and the mentioned trade-offs are evaluated in experiments on data sets from the UCI Machine Learning Repository [23].

Strechtt, Moreira and Soares [2] research on educational data mining starts from the premise that predicting the failure of students in university courses can provide useful information for course and programme managers as well as to explain the drop out phenomenon. The rationale is that while it is important to have models at course level, their number makes it hard to extract knowledge that can be useful at the university level. Therefore, to support decision making at this level, it is important to generalize the knowledge contained in those models. An approach is presented to group and merge interpretable models in order to replace them with more general ones without compromising the quality of predictive performance. The case study is data from the University of Porto, Portugal, which is used for evaluation. The aggregation method consists mainly of intersecting the decision rules of pairs of models of a group recursively, i.e., by adding models along the merging process to previously merged ones. The results obtained are promising, although they suggest alternative approaches to the problem. Decision trees were trained using C5.0 algorithm and F1 was used as evaluation function of the individual and merged models.
3 Definition of the problem

Many classification (i.e. here decision tree) and prediction methods have been proposed by researcher in machine learning pattern recognition and statistics for distributed environment, and they also have proposed different approaches for merging the local decision trees. From the deep literature review the facts have been identified that, most algorithms are memory resident, typically assuming a small data size, not domain-free, static in nature, less efficient in terms of processing and communication overhead. Due to the large volume of data with privacy concern there should be some efficient technique which supports scalable and dynamic classification and prediction capable of handling large distributed data sets and generates the global decision tree without losing the prediction quality.

Based on the state of the art, we come to the research problem of designing the efficient technique for merging the decision trees which supports scalable and dynamic classification in distributed environment with large volume of data.

4 Objective and scope of work

OBJECTIVE

A series of challenges have recently emerged in the data mining field for distributed environment, triggered by the rapid shift in status from academic to applied science and the resulting needs of real-life applications. The proposed work is concerned with dynamic and scalable approach for merging the decision trees models for large volume of data in distributed environment. In the following, the main objectives of the thesis are enlisted.

1) To reduce the model (i.e. decision tree) training time and communication time in distributed environment for large volume of data.
2) To introduce the efficient scalable and dynamic approach for newly generated dataset and already trained model.
3) To prepare the rule merging policies to generate the global model.
4) To generate the globally interpretable model by preserving the prediction quality.

SCOPE

In this research the following things have been considered/included as the scope.

1) To work with homogeneous and horizontally fragmented data set.
2) To collect and pre-process real data set: The student admission data set from Official Web Portal of Parul group of institutes.
3) The work has been carried out on educational data and mainly has been focused on student admission prediction.
4) The parser has been proposed to convert the decision tree into the decision rule.
5) The outcome of this research at the end is optimized global decision tree without losing the prediction quality.
6) The simulation of work has been carried out on 2, 5 and 10 sites in the network.
5 Original contributions by the thesis

The entire work in this synopsis, as well as thesis is the original work, with the copyright and the research papers as the back bone. The proposed framework and the algorithms have been visualized as a collection of various modules, each of which with relevant publications. The details of the associated copyright and papers are as follows:

Copyright Applied:

Paper Presented / Published: Total 8 papers in national/international journals/conferences

Book Chapter Approved:
Book Title: Web Usage Mining Techniques and Applications Across Industries
Book Editor: Dr. A.V. Senthil
Chapter Title: A Dynamic and Scalable Decision Tree Based Mining of Educational Data
Publication: IGI Global International

Book/Paper Submitted: Total 3 papers in International journal/conference are in process and one book is under review for publication.

6 Methodology of research, results / comparisons

6.1 Methodology of research
In this research work the qualitative and exploratory approaches have been used and followed the research methodology steps. Very first, during the literature review we referred various research papers, patents and other articles on dynamic and scalable data mining for distributed environment, classification techniques and algorithms, merging the decision trees and educational data mining. In addition to this, we installed Weka tool [17] which is an open source by the University of Waikato of New Zealand and studied various supervised data mining algorithms for classification. During this initial phase of literature review, we found researchers had done work on classification algorithms but very few of them worked on decision tree in distributed environment. Major reasons for this gap are most of the algorithms are not memory resident, static in nature, scalable and domain free and also very less work have been done for dynamic and scalable distributed environment.

During our literature review, we found research works based on merging the decision trees generated at different geographical locations to form global decision tree without losing the
predictive quality. Therefore, our second phase of literature review was mainly focused on global model generation from different local decision trees. By studying and comparing various approaches, we found merging the decision rules are challenging task. We also found none of the researcher has worked on scalable and dynamic approach in distributed environment. As a result of both phases of literature review, we proposed a model (with framework, system architectures and algorithms) with the objectives 1) to reduce network overhead, 2) Scalable and dynamic distributed environment support which let not need to process whole dataset every time and 3) The global model should not loss the predictive quality. The proposed model with the frame work, system architecture at local site, decision tree merging architecture at coordinator site are shown in figures 1,2 and 3 respectively. The detail of each is available in the further sub sections.

To fulfill the objectives, the proposed model has been implemented in two phases. In the first phase, 1) the decision trees have been generated at each local site, 2) the decision tables have been formed from each local site, 3) conversion of each local decision table into XML file to transmit it over the internet. In the second phase, 1) the XML files have been converted into the decision tables, 2) all decision tables have been merged and 3) the resultant decision table have been converted into XML file to send to all local sites for prediction.

The proposed model has been implemented on the educational data set. We have used the real data set of student admission process in different disciplines. The data sets have been collected from Parul university web portal (PUWP). In the experimental the data set is processed on 2, 5 and 10 different sites to generate the local decision tree models which later merged into a single decision tree as a whole without losing the predication quality

6.2 The model and its components

The proposed framework, the system architecture at local site, the system architecture at coordinator site and the rule merging policies have been discussed in the following subsections.

6.2.1 The proposed framework

As shown in figure-1, the data set D as a whole considered partitioned across different data set sites Si where i=1,2,3,…d each site Si now process the locally available dataset Di to generate the decision tree using C5.0 algorithm in weka© tool.
C4.5 is an algorithm used to generate a decision tree developed by Ross Quinlan. C4.5 is an extension of Quinlan's earlier ID3 algorithm. The decision trees generated by C4.5 can be used for classification, and for this reason, C4.5 is often referred to as a statistical classifier. C4.5 builds decision trees from a set of training data in the same way as ID3, using the concept of information entropy. The training data is a set \( S = S_1, S_2, \ldots \) of already classified samples. Each sample \( S_i = X_1, X_2, \ldots \) is a vector where \( X_1, X_2, \ldots \) represent attributes or features of the sample. The training data is augmented with a vector \( C = C_1, C_2, \ldots \) Where \( C_1, C_2, \) represent the class to which each sample belongs. At each node of the tree, C4.5 chooses one attribute of the data that most effectively splits its set of samples into subsets enriched in one class or the other. Its criterion is the normalized information gain (difference in entropy) that results from choosing an attribute for splitting the data. The attribute with the highest normalized information gain is chosen to make the decision. This process will continue and will form the decision tree.

As the decision trees generated at each site occupies larger memory, hence it is converted into decision tables followed by XML files to transmit over the network such that very less network overhead takes place. Each XML file then later available at coordinator site, where actual decision tree merging process takes place.

The following figure 2 and figure 3 shows the flow chart of the complete flow of our experimental algorithm at local site and coordinator site respectively.
START
Site Si, Dataset Di, New Dataset at each site Dtsi
For i=1,2,3,……..,N
Flag=0 (Dataset not processed before) i ← 0

If flag=0
M←Cardinality(Dtsi)
j←0

Generate Decision tree DTi using C5.0
Generate Decision tree DTi using C5.0
Convert DTi into decision table Dtablei.
flag←1
Set Decision rules Dtablei in Desc. Order as
class label majority
Create XML file Xi for Dtablei
Send Xi to coordinator site

If tuple tij doesn’t follow Decision Tree rules
Create new rule

count ← count(rule)
Increment count & classify the rule

If tuple tij doesn’t follow most of(except one) Decision Tree rules

If tuple tij conflicts any rule
Ignore it

If tuple tij partially or fully overlapped by the Decision Tree rule?

Modify the D.T accordingly

If i>N
STOP

Yes

No

Yes

Yes

Yes

No

No

No

No

No

j←j+1

Create XML file Xi for Dtablei
Send Xi to coordinator site

j←j+1

i←i+1

Figure 2: Local site processing

13
6.2.2 The system architecture at local site

As shown in the figure-4 below, the detailed proposed architecture for dynamic and scalable decision tree generation process has been discussed. The very first step is model Mi creation from the data set Di available at each site. Later the parser converts the decision trees into the decision rule set Ri for each site Si. In the third phase the decision rule set Ri is converted into decision table DTablei which later converted into XML file.
Each local site $S_i$ sends its locally generated XML files $X_i$ to coordinator site for further decision tree merging process. In one of the intermediate step, the newly added data set is appended with the previous decision table $D_{tablei}$ of site $S_i$ directly generating the decision tree of new data set. This way the approach becomes scalable, i.e. the algorithm supports new data sets as well.

### 6.2.3 The system architecture at coordinator site

The process of merging $k$ decision trees $F_1, F_2, F_3, \ldots F_k$ into a single one starts with creating for each tree $F_i$ its Decision Table set $D_{table}(F_i)$. Decision Tables $D_{table}(F_1), D_{table}(F_2), \ldots$ is reduced into a final Decision Table $D_{table\_Final}$ by the merging operation on Decision tables set. Finally, $D_{table\_Final}$ is turned into a decision tree. The merging operation is the core of the approach: it merges several decision table sets $D_{table}(F_1), D_{table}(F_2), \ldots$ etc into a single one. It consists of several phases like intersection, filtering and reduction as shown in figure-5 below.
As shown in figure 5 above, the decision tables Dtable(Fi) of all the sites Si with data set Di where i=1,2,3,…d are merged. At very first the intersection phase is carried out where the common regions i.e. rules are found. In the second phase, the less useful disjoint regions are removed from the list. This process is known as filtering. In the third reduction phase, the disjoint regions which can be combined i.e. merged with minor changes are merged to reduce the number of disjoint regions.

**Intersection Phase:** It is a task to combine the regions of two decision models using a specific method to extract the common components of both, presented in decision table. The set of values (Numerical only) of each region on each model are compared to discover common sets of values across each variable. The class to assign to the merged region is straightforward if the pair of regions have the same class, otherwise class conflict problem arises. Andrzejak, Langer and Zabala[7] propose three strategies to address this problem. a) Assign the class with the greatest confidence, b) Assign the class with the greater probability c) Retrain the model with examples for conflicting class regions, If no conflict arises that class is assigned. Otherwise remove that region from the merged model.

**Filter Phase:** It is the task to remove the disjoint regions from the intersected model. This is some what pruning operation. In this the regions with the highest relative volume and number of training examples are retained. Strecht, Moreira and Soares [6] address the issue by removing the disjoint regions, and highlighted the case where the models are not mergeable if all regions are disjoint.

**Reduction Phase:** This is applicable when a set of regions have the same class and all variables have equal values except for one. To obtain the simpler merged model, this is the task to find out which can be joined into one. For Nominal Variables: Union of values of variables from all regions. For Numeric Variables: If intervals are contiguous.

### 6.2.4 Decision rule merging policies

**Rule-1:** The continuous value should not be differing by any more than threshold (this is adjustable).

**Rule-2:**

- **Rule-2A:** If the attribute tests “>” then the smaller of the two rule values is used.
- **Rule-2B:** If the attribute tests “<=“then the larger of the two rule values is used.

**Rule-3:** Partial Overlap

Two rules in which conditions are partially overlap; adjust the boundaries of the rule

**Rule-4:** One rule completely overlaps other rule, modify overlapped rule according to overlapping rule.

**Rule-5:** Conflict in Label

Two same rules have different labels then select the rule as below

- a) Use the label with highest confidence
- b) Avg. the probability distribution and use the label with highest probability
6.3 Results / Comparisons

<table>
<thead>
<tr>
<th></th>
<th>Accuracy</th>
<th>Error Rate</th>
<th>Sensitivity</th>
<th>Specificity</th>
<th>Precision</th>
<th>Recall</th>
<th>Training Time (Sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SITE1</td>
<td>90.845</td>
<td>9.115</td>
<td>100</td>
<td>74</td>
<td>94.63</td>
<td>0.789</td>
<td>0.11</td>
</tr>
<tr>
<td>SITE2</td>
<td>95.53</td>
<td>4.47</td>
<td>100</td>
<td>78.94</td>
<td>87.62</td>
<td>0.648</td>
<td>0.03</td>
</tr>
<tr>
<td>COMBINED</td>
<td>98.13</td>
<td>1.87</td>
<td>100</td>
<td>93.75</td>
<td>97.40</td>
<td>0.701</td>
<td>0.05</td>
</tr>
</tbody>
</table>

Table 3: Performance measure of the proposed algorithm for student admission data set

![Performance comparison for different parameters](image1)

Figure 6: Performance comparison for different parameters

<table>
<thead>
<tr>
<th></th>
<th>No. Of Instances</th>
<th>Accuracy</th>
<th>Error Rate</th>
<th>Sensitivity</th>
<th>Specificity</th>
<th>Precision</th>
<th>Recall</th>
<th>Training Time (Sec)</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Student Admission</td>
<td>179</td>
<td>98.13</td>
<td>1.87</td>
<td>100</td>
<td>93.75</td>
<td>97.40</td>
<td>98.13</td>
<td>0.05</td>
<td></td>
</tr>
<tr>
<td>Student Performance</td>
<td>100</td>
<td>93</td>
<td>7</td>
<td>100</td>
<td>92.63</td>
<td>93.2</td>
<td>93</td>
<td>0.02</td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Student admission and student performance dataset with different parameters
Comparison of proposed approach with other approaches

Table 5: Performance of three approaches with 2 sites

<table>
<thead>
<tr>
<th>Records</th>
<th>Centralized Approach</th>
<th>Intermediate Message Passing</th>
<th>Proposed Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>10000</td>
<td>17.7</td>
<td>11.021</td>
<td>6.4</td>
</tr>
<tr>
<td>20000</td>
<td>40.8</td>
<td>25.83</td>
<td>14.3</td>
</tr>
<tr>
<td>40000</td>
<td>102.1</td>
<td>59.79</td>
<td>29.32</td>
</tr>
<tr>
<td>60000</td>
<td>130.6</td>
<td>97.59</td>
<td>36</td>
</tr>
<tr>
<td>80000</td>
<td>205.7</td>
<td>128.98</td>
<td>58.6</td>
</tr>
<tr>
<td>100000</td>
<td>229.3</td>
<td>160.33</td>
<td>70.73</td>
</tr>
</tbody>
</table>

Figure 7: Performance comparison on two different datasets

Figure 8: Performance comparison among centralized, intermediate message passing and proposed approach for 2 sites

Table 6: Performance of three approaches with 5 sites

<table>
<thead>
<tr>
<th>Records</th>
<th>Centralized Approach</th>
<th>Intermediate Message Passing</th>
<th>Proposed Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>10000</td>
<td>14.4</td>
<td>10.78</td>
<td>3.33</td>
</tr>
<tr>
<td>20000</td>
<td>34.79</td>
<td>29.77</td>
<td>7.61</td>
</tr>
<tr>
<td>40000</td>
<td>73.36</td>
<td>59.83</td>
<td>16.91</td>
</tr>
<tr>
<td>60000</td>
<td>92.06</td>
<td>80.72</td>
<td>21.65</td>
</tr>
<tr>
<td>80000</td>
<td>150.97</td>
<td>116.03</td>
<td>32.4</td>
</tr>
<tr>
<td>100000</td>
<td>164.24</td>
<td>149.48</td>
<td>38.05</td>
</tr>
</tbody>
</table>
Figure 9: Performance comparison among centralized, intermediate message passing and proposed approach for 5 sites

<table>
<thead>
<tr>
<th>Records</th>
<th>Centralized Approach</th>
<th>Intermediate Message Passing</th>
<th>Proposed Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>10000</td>
<td>8.33</td>
<td>5.78</td>
<td>3.02</td>
</tr>
<tr>
<td>20000</td>
<td>17.01</td>
<td>15.08</td>
<td>6.28</td>
</tr>
<tr>
<td>40000</td>
<td>35.54</td>
<td>30.49</td>
<td>14.903</td>
</tr>
<tr>
<td>60000</td>
<td>46.46</td>
<td>45.09</td>
<td>18.07</td>
</tr>
<tr>
<td>80000</td>
<td>77.61</td>
<td>63.71</td>
<td>26.48</td>
</tr>
<tr>
<td>100000</td>
<td>98.331</td>
<td>73.29</td>
<td>31.31</td>
</tr>
</tbody>
</table>

Table 7: Performance of three approaches with 10 sites

Figure 10: Performance comparison among centralized, intermediate message passing and proposed approach for 10 sites

7 Achievements with respect to objectives
The outcome of the proposed model shows that the objectives of the research work have been acquired. The proposed model can handle large volume of data. The decision trees are merged with minimal network overhead and global model preserves the quality of prediction. The results shown in the above tables with accuracy, error rate, specificity, sensitivity, precision, recall and training time are better than the existing system. The total time for local model generation and communication time in proposed approach is ~3.14 and ~2.34 times
faster than the centralized and intermediate message passing approaches. The data set of student admission for the year 2013-14, 2014-15 have been used to train the model, this model has been used with the data set of student admission for the year 2015-16 which gives more than 98.03% accuracy for the prediction. These experimental results have been also verified using the 10-fold cross validation.

8 Conclusions

Decision tree learning on massive datasets is a common data mining task in distributed environment, yet many state of the art tree learning algorithms require training data to reside in memory on a single machine, while more scalable implementations of tree learning have been proposed, they typically require specialized parallel computing architectures. Moreover, all the approaches are static in nature, not domain-free, not scalable and the accuracy is not preserved.

Our literature review and experiments on merging the decision trees, shown training time, communication overhead and the accuracy are the major challenges. To reduce the training time the proposed algorithm processes only new dataset with already trained model which makes it scalable and dynamic, to reduce the communication overhead the local models have been converted into XML files, to preserve the accuracy the proposed algorithm incorporates some rule merging policies.

The proposed approach has been divided into two major phases. During first phase, the objectives were a) minimize the training time and b) reduce the communication overhead by the sub-phases 1) Generating the local decision tree model, 2) Applying the parsing for converting decision tree into decision table, 3) Converting this decision table into XML file to reduce the communication overhead and 4) Applying the scalable approach with new data set. The total time for local model generation and communication time in proposed approach is ~3.14 and ~2.34 times faster than the centralized and intermediate message passing approaches.

During the second phase of proposed approach, we have introduced several rule merging policies to preserve the quality by performing model intersection, filtering and reduction phases. We have compared the resultant merged and trained model with the actual one generated by C5.0 algorithm. This model preserves the accuracy.

We proposed a scalable and dynamic distributed approach for learning tree models over large datasets which defines tree learning as a series of distributed computations. We show how this approach supports dynamic and scalable construction of decision trees models, as well as ensembles of such models. The proposed approach is much efficient than all others.
9 Copies of papers published and a list of all publications arising from the thesis

9.1 Paper presented / published

1) Students' Admission Prediction using GRBST with Distributed Data Mining, Communications on Applied Electronics (CAE) – ISSN : 2394-4714, Foundation of Computer Science FCS, New York, USA, Volume 2 – No.1, June 2015
2) A Proposed DDM Algorithm and Framework For EDM of Gujarat Technological University, Organized by Saffrony Institute of Technology International Conference on Advances in Engineering, 22nd-23rd January 2015
3) An Approach on Early Prediction of Students’ Performance in University Examination of Engineering Students Using Data Mining, International Journal of Scientific Research and Management Studies (IJSRMS) ISSN: 2349-3371 Volume 1 Issue 5, pg: 156-161
4) Faculty Performance Evaluation Based on Prediction in Distributed Data Mining, 2015 IEEE ICETECH- Coimbatore
5) Prediction and analysis of student performance using distributed data mining, International Conference on Information, Knowledge & Research In Engineering, Management and Sciences(IC-IKR-EMS), 7th Dec-2014 KIT, Gujarat. IJETAETS-ISSN 0974-3588
6) Prediction and analysis of Faculty performance using distributed data mining, International Conference on Information, Knowledge & Research In Engineering, Management and Sciences(IC-IKR-EMS), 7th Dec-2014 KIT, Gujarat. IJETAETS-ISSN 0974-3588
7) A decision support application for student admission process based on prediction in distributed data mining, International Conference on Information, Knowledge & Research In Engineering, Management and Sciences(IC-IKR-EMS), 7th Dec-2014 KIT, Gujarat. IJETAETS-ISSN 0974-3588
8) A Dynamic and Scalable Evolutionary Data Mining for Distributed Environments, NCEVT-2013, PIET, Limda

9.2 Papers in communication:
1) Students’ admission prediction using GRBST with distributed data mining, ACM Transaction on Knowledge Discovery in Data, Impact Factor-0.93
2) Dynamic and Scalable Data Mining with an Incremental Decision Trees Merging Approach for Distributed Environment, A Doctoral Conference 2016 (DocCon 2016) at Udaipur, March-2016. (Under Publication)
3) An Approach of E-Governance with Distributed Data Mining For Student Performance Prediction, Springer international conference, ICICT-October-2015, Udaipur. (Under Publication)
10 Patents/Copyright (if any)

<table>
<thead>
<tr>
<th>Title</th>
<th>A DYNAMIC AND SCALABLE EVOLUTIONARY DATA MINING &amp; KDD FOR DISTRIBUTED ENVIRONMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filed At</td>
<td>Copyright Office, New Delhi</td>
</tr>
<tr>
<td>Application No.</td>
<td>Dairy Number:3460/2016-CO/SW</td>
</tr>
<tr>
<td>Application Date</td>
<td>18/03/2016</td>
</tr>
<tr>
<td>Applicants &amp; Inventors</td>
<td>Dineshkumar Bhagwandas Vaghela</td>
</tr>
<tr>
<td></td>
<td>Dr. Priyanka Sharma</td>
</tr>
<tr>
<td>Application Status</td>
<td>In Process</td>
</tr>
<tr>
<td>Objection Received</td>
<td>Not Yet</td>
</tr>
</tbody>
</table>

11 Achievements

- Published/presented 8 papers in national/international level journals/conferences and three papers in process.
- One book chapter has been selected in book titled “Web Usage Mining Techniques and Applications across Industries” by Dr. A.V. Senthil with IGI global international publication.
- The papers have been cited in 10 papers and the H index is 2.
- Book on data mining is under review for publication.

12 Partial Lists of References


Baik, S. Bala, J. (2004). A Decision Tree Algorithm For Distributed Data Mining.

http://www.cs.waikato.ac.nz/ml/weka/


