ACCURATE DECISION TREE CONSTRUCTION

C. SUDARSANA REDDY[1]
Department of Computer Science and Engineering,
S.V. University College of Engineering,
S.V. University,
Tirupati, Andhra Pradesh, India
Email: cheruku1sudarsana2reddy3@gmail.com

J. NAGA MUNEIAH[2]
Associate professor of Computer science and Engg
Department of Computer Science and Engineering,
Chadalawada Ramanamma Engineering College
Tirupati, Andhra Pradesh, India

S. AQUTER BABU[3]
Assistant Professor of Computer Science
Department of Computer Science
Dravidian University
Kuppam -517425
Chittoor District, Andhra Pradesh, India
s_a_babu1@yahoo.co.in
09940263687

Abstract— Classification is one of the most important techniques in data mining. Decision tree is the most important classification technique in machine learning and data mining. Decision tree classifiers are constructed using training dada sets. Training data sets contain numerical (or continuous) and categorical (or discrete) attributes. Measurement errors are common in any data collection process, particularly when training datasets contain numerical (or continuous) attributes. So, values of numerical attributes contain measurement errors in many training data sets.

We extend certain (or traditional or classical) decision tree building algorithms to handle values of numerical attributes containing measurement errors. We have discovered that the accuracy of a certain (or classical or traditional) decision tree classifiers can be much improved if the measurement errors in the values of numerical attributes in the training data sets are properly handled (or controlled or modeled or corrected) appropriately.

The present study proposes a new algorithm for decision tree classifier construction. This new algorithm is named as Accurate Decision Tree (ADT) classifier construction. ADT classifiers are more accurate than certain (or traditional or classical) decision tree classifiers. ADT proves to be more effective regarding classification accuracy when compared with Certain Decision Tree (CDT) classifiers. The performance of these two algorithms is compared experimentally through simulation.

Keywords-error corrected values of the numerical attributes in the training data sets; measurement errors in the values of numerical attributes in the training data sets; training data sets containing numerical attributes; training data sets, decision tree; classification.

I. INTRODUCTION

Decision tree is the most commonly used data classification technique. Decision tree induction is the learning of decision trees from class labeled training tuples [1]. Two most important features of decision tree are comprehensibility and interpretability [1]. In general, training data sets contain both numerical (continuous) and categorical (discrete) attributes. Raw measured data values of numerical attributes normally contain measurement errors. A new decision tree classifier construction method based on the error correcting idea constructs more accurate decision tree classifiers.

In traditional or classical decision tree classification, decision tree classifiers are constructed directly from the values of the attributes of the training data sets without considering data measurement errors present in the values of numerical attributes in the training data sets. We call this approach certain decision tree (CDT). Another approach during the decision tree classifier construction is to consider the data errors present in the values of numerical attributes of training data sets. We call this approach Accurate Decision Tree (ADT)

classifier construction method. When decision trees are used for classification they are called classification trees [2]. Decision tree classifiers are popular because they learn and respond quickly and accurate in many domains [2].

In this paper a new decision tree classifier construction algorithm is proposed. The new decision tree classifier construction algorithm, ADT, takes care of measurement errors present in the values of numerical attributes in the training data sets. Accurate decision tree (ADT) classifier construction method can build significantly more accurate decision trees than certain decision tree (CDT) classifier construction methods. High classification accuracies can be achieved by using accurate decision trees (ADTs).

One of the most popular classification models is the decision tree model [3]. Accurate decision tree (ADT) construction method can potentially build more accurate decision tree classifiers because it takes measurement error information into account.

We cannot always assume that the training data sets are error free [3]. It is likely that some sort of measurement errors are incurred in the collection process of these training data sets [3]. Errors may occur in random fashion. Sometimes the errors in the values of the numerical attributes in the training data sets can be modeled using statistical distributions such as Gaussian and Uniform distributions. In the case of random noise better to use Gaussian distribution to model errors present in the values of the numerical attributes in the training data sets. Many data sets with numerical attributes have been collected via repeated measurements and the process of repeated measurements is the common potential source of getting measurement errors in the values of numerical attributes in the training data sets. Sometimes values of numerical attributes in the training data sets are collected over an unspecified number of repeated measurements [3].

Data obtained from measurements by physical devices are often inaccurate due to measurement errors [3]. Another source of error is quantization errors introduced by the digitization process [3]. Such errors can be properly handled by assuming an appropriate error correcting model such as Gaussian error distribution for random noise or a uniform error distribution for quantization errors [3].

In general, errors play an important role in every scientific and medical experiment. There exists many data errors and random errors are the most important data errors to be considered in scientific and medical experiments. Present study mainly concentrates to find and correct random errors present in the values of numerical attributes in the training dada sets by systematically adjusting various random data error values in the numerical attributes of the training dada sets. It is possible to build decision tree classifiers with higher accuracies especially when measurement errors are modeled appropriately.

Decision trees have been well recognized as very powerful and attractive classification tools [4]. Errors in scientific experiments are extremely well approximated by a normal distribution [5]. Normal distribution equation is also derived from a study of errors in repeated measurements of the same quantity [5]. The term continuous is used in the literature to indicate both real and integer valued attributes [8]. A classification rule will be expressed as a decision tree [9].

II. PROBLEM DEFINITION

In many real life applications training data sets are not error free due to measurement errors in data collection process. In general, values of numerical attributes in training data sets are always inherently associated with errors. Different types of errors present in the training data sets are not considered during decision tree construction of existing decision tree classifiers. Hence, classification results of existing decision tree classifiers are less accurate or inaccurate in many cases because of different types of data errors present in the numerical attributes of the training data sets.

As data errors are associated with almost all training data sets containing numerical attributes, it is important to develop more accurate data mining techniques by considering error corrected data values of numerical attributes of the training data sets.

Sometimes, for preserving data privacy training data sets are modified explicitly by adding certain data error values into the values of numerical attributes in training data sets. So, in such cases training data sets contain errors with modified attribute values. Such modified data sets must be reconstructed by eliminating explicitly added data errors into the training data sets.

III. EXISTING ALGORITHM

A. Certain Decision Tree (CDT) Algorithm Description

The certain decision tree (CDT) algorithm constructs a decision tree classifier by splitting each node into left and right nodes. Initially, the root node contains all the training tuples. The process of partitioning the training data tuples in a node into two subsets based on the best split point value z_T of best split attribute A_{j_T} and storing the resulting tuples in its left and right nodes is referred to as splitting. Whenever further split of a node is not required then it becomes a leaf node referred to as an external node. All other nodes except root node are referred as internal nodes. The splitting process at each internal node is carried out recursively until no further

split is required. Continuous valued attributes must be discretized prior to attribute selection [6]. Further splitting of an internal node is stopped if one of the stopping criteria given hereunder is met.

1. All the tuples in an internal node have the same class label. 2. Splitting does not result nonempty left and right nodes.

In the first case, the probability for that class label is set to 1 whereas in the second case, the internal node becomes external node. The empirical probabilities are computed for all the class labels of that node. The best split pair comprising an attribute and its value is that associated with minimum entropy.

Entropy is a metric or function that is used to find the degree of dispersion of training data tuples in a node. In decision tree construction the goodness of a split is quantified by an impurity measure [2]. One possible function to measure impurity is entropy [2]. Entropy is an information based measure and it is based only on the proportions of tuples of each class in the training data set. Entropy is used for finding how much information content is there in a given data [1].

Entropy is taken as dispersion measure because it is predominantly used for constructing decision trees. In most of the cases, entropy finds the best split and balanced node sizes after split in such a way that both left and right nodes are as much pure as possible.

Accuracy and execution time of CDT algorithm for 9 data sets are shown in Table 5.2

Entropy is calculated using the formula

$$entropy(S) = \sum_{i=1}^{m} -p_i . log_2(p_i)$$

Where p_i = number of tuples belongs to the i^{th} class

$$H(z, A_j) = \sum_{X=L,R} \frac{|X|}{|S|} \left(\sum_{c \in C} -\frac{p_c}{X} log_2 \left(\frac{p_c}{X} \right) \right)$$

$$H(z, A_j) = \frac{|L|}{|S|} \left(\sum_{c \in C} -\frac{p_c}{L} log_2 \left(\frac{p_c}{L} \right) \right) + \frac{|R|}{|S|} \left(\sum_{c \in C} -\frac{p_c}{R} log_2 \left(\frac{p_c}{R} \right) \right) \quad (3.1)$$

$$H(z, A_j) = \frac{|L|}{|S|} \left(Entropy(L) \right) + \frac{|R|}{|S|} \left(Entropy(R) \right)$$

Where

- A_j is the splitting attribute.
- L is the total number of tuples to the left side of the split point z.
- R is the total number of tuples to the right side of the split point z.
- $\frac{p_c}{L}$ is the number of tuples belongs to the class label c to the left side of the split point z.
- $\frac{p_c}{R}$ is the number of tuples belongs to the class label c to the right side of the split point z.
- S is the total number of tuples in the node.

B. Pseudo code for Certain Decision Tree (CDT) Algorithm

```
CERTAIN_DECISION_TREE (T)
1.  If all the training tuples in the node T have the same class label then
2.      set p_T(c) = 1.0
3.  return(T)
4.  If tuples in the node T have more than one class then
5.  Find_Best_Split(T)
6.  For i ← 1 to datasize[T] do
7.  If split_atribute_value[t_i] <= split_point[T] then
8.      Add tuple t_i to left[T]
9.  Else
10.     Add tuple t_i to right[T]
11. If left[T] = NIL or right[T] = NIL then
12.     Create empirical probability distribution of the node T
13.     return(T)
14. If left[T] != NIL and right[T] != NIL then
15.     CERTAIN_DECISION_TREE(left[T])
```

16. CERTAIN_DECISION_TREE(right[T])
17. return(T)

IV. PROPOSED ALGORITHM

A. Proposed Accurate Decision Tree (ADT) Algorithm Description

The procedure for creating accurate decision tree (ADT) classifier is same as that of certain decision tree (CDT) classifier construction except that ADT calculates entropy values for error corrected data values in the numerical attributes of the training data sets. Errors in the values of numerical attributes in the training datasets are calculated based on the assumption that training data sets contain measurement errors particularly when the training data sets contain numerical attributes.

Based on the assumption that measurement errors are inevitable in the values of numerical attributes in the training data sets, errors are corrected in the values of numerical attributes by assuming 1% or 0.1% or 0.01% errors in the values of numerical attributes and then entropy is calculated for each value of each attribute in the training data set. Extensive simulation experiments have been conducted which show that the resulting experiments are more accurate than those of certain decision trees (CDT). ADT can build not only more accurate decision tree classifier but also it is more efficient than CDT. Execution times of both the algorithms are same.

The present study has verified experimentally through simulation the performance of two algorithms, Certain Decision Tree (CDT) and Accurate Decision Tree (ADT). In this paper we have constructed decision tree classifiers with training data sets containing only numerical (or continuous) attributes having data measurement errors. Later on we will extend the procedure for constructing decision tree classifiers with training data sets containing both numerical (continuous) and categorical (discrete) attributes also. In real life many training data sets may contain errors other than data measurement errors. Such errors must be controlled appropriately in order to construct more accurate decision tree classifiers. Also some pruning techniques are needed to improve the performance.

Accuracy and execution time of CDT algorithm for 9 data sets are shown in Table 5.2. Accuracy and execution time of ADT algorithm for 9 data sets are shown in Table 5.3 and comparison of execution time and accuracy for CDT and ADT algorithms for 9 data sets are shown in Table 5.4 and charted in Figure 5.1 and Figure 5.2 respectively.

B. Pseudo code for Accurate Decision Tree (ADT) Algorithm

ACCURATE_DECISION_TREE (T)

1. If all the training tuples in the node T have the same class label then
2. set $p_T(c) = 1.0$
3. return(T)
4. If tuples in the node T have more than one class then
5. **For each value of each numerical attribute first correct data error and then find entropy.**
6. Find_Best_Split(T)
7. For i ← 1 to datasize[T] do
8. If split_atribute_value[t_i] <= split_point[T] then
9. Add tuple t_i to left[T]
10. Else
11. Add tuple t_i to right[T]
12. If left[T] = NIL or right[T] = NIL then
13. Create empirical probability distribution of the node T
14. return(T)
15. If left[T] != NIL and right[T] != NIL then
16. ACCURATE_DECISION_TREE(left[T])
17. ACCURATE_DECISION_TREE(right[T])
18. return(T)

V. EXPERIMENTAL RESULTS

A simulation model is developed for evaluating the performance of two algorithms: Certain Decision Tree (CDT) and Accurate Decision Tree (ADT) experimentally. The data sets shown in Table 5.1 from University of California (UCI) Machine Learning Repository are employed for evaluating the performance of the above said algorithms.

In all our experiments we have used data sets from the UCI Machine Learning Repository [7]. 10-fold cross-validation technique is used for test tuples for all training data sets with numerical attributes except Satellite and PenDigits training data sets [7]. For Satellite and PenDigits training data sets with numerical attributes a separate test data set is used for testing.

The simulation model is implemented in Java 1.7 on a Personal Computer with 3.22 GHz Pentium Dual Core processor (CPU), and 2 GB of main memory (RAM). The performance measures, accuracy and execution time, for the above said algorithms are presented in TABLE 5.2 to TABLE 5.4 and Fig 5.1 to Fig 5.4.

Table 5.1 Data Sets from the UCI Machine Learning Repository

No	Data Set Name	Training Tuples	No.Of Attributes	No. Of Classes	Test Tuples
1	Iris	150	4	3	10-fold
2	Glass	214	9	6	10-fold
3	Ionosphere	351	32	2	10-fold
4	Breast	569	30	2	10-fold
5	Vehicle	846	18	4	10-fold
6	Segment	2310	14	7	10-fold
7	Satellite	4435	36	6	2000
8	Page	5473	10	5	10-fold
9	Pen Digits	7494	16	10	3498

TABLE 5.2 Certain Decision Tree (CDT) Accuracy and Execution Time

No.	**Data Set Name**	**Total Tuples**	**Accuracy**	**Execution Time**
1	Iris	150	98.0	1.0
2	Glass	214	90.9524	1.2
3	Ionosphere	351	82.2857	1.037
4	Breast	569	95.0969	2.224
5	Vehicle	846	78.6905	5.63
6	Segment	2310	94.4156	27.524
7	Satellite	4435	83.3999	145.308
8	Page	5473	98.5558	46.374
9	Pen Digits	7494	91.0234	640.03

TABLE 5.3 Accurate Decision Tree (ADT) Accuracy and Execution Time

No	**Data Set Name**	**Total Tuples**	**Accuracy**	**Execution Time**	**Error value**
1	Iris	150	98.6667	1.0	0.01
2	Glass	214	95.7943	1.2	0.01
3	Ionosphere	351	97.1429	2.208	0.001
4	Breast	569	95.8929	2.543	0.0001
5	Vehicle	846	85.9524	5.856	0.0001
6	Segment	2310	95.368	28.971	0.0001
7	Satellite	4435	85.2	144.129	0.01
8	Page	5473	98.6106	43.486	0.001
9	Pen Digits	7494	91.9319	639.03	0.1

TABLE 5.4 Comparison of accuracy and execution times of CDT and ADT

No.	**Data Set Name**	**CDT Accuracy**	**ADT Accuracy**	**CDT Execution Time**	**ADT Execution Time**
1	Iris	98.0	98.6667	1.0	1.0
2	Glass	90.9524	95.7943	1.2	1.2
3	Ionosphere	82.2857	97.1429	1.037	2.208
4	Breast	95.0969	95.8929	2.224	2.543
5	Vehicle	78.6905	85.9524	5.63	5.856
6	Segment	94.4156	95.368	27.524	28.971
7	Satellite	83.3999	85.2	145.308	144.129
8	Page	98.5558	98.6106	46.374	43.486
9	Pen Digits	91.0234	91.9319	640.03	639.03

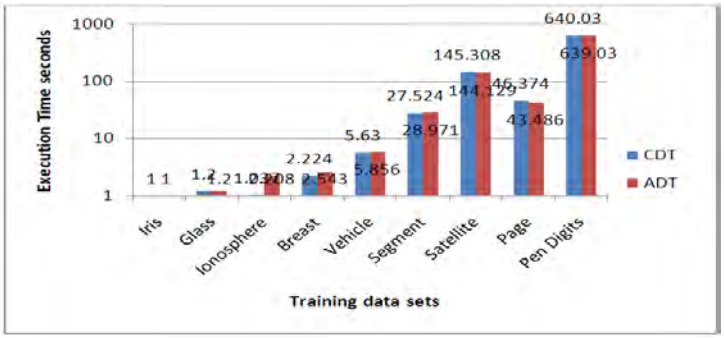

Fig 5.1 Comparison of execution times of CDT and ADT

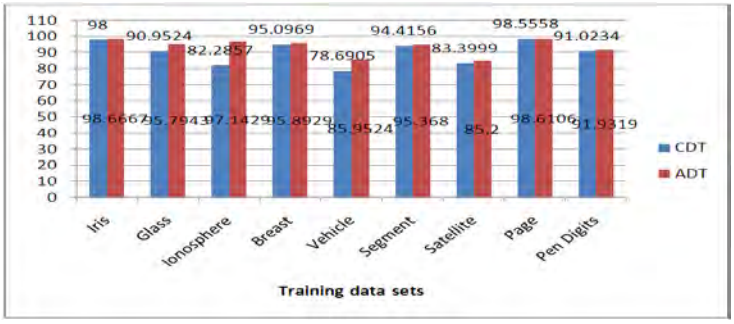

Fig 5.2 Comparison of Classification Accuracies of CDT and ADT

VI. CONCLUSIONS

A. Contributions

The performance of existing traditional or classical or Certain Decision Tree (CDT) is verified experimentally through simulation. A new decision tree classifier construction algorithm called Accurate Decision Tree (ADT) is proposed and compared with the existing certain decision tree classifier. It is experimentally found that the classification accuracy and performance of proposed algorithm (ADT) is much better than CDT algorithm.

B. Limitations

Proposed algorithm, Accurate Decision Tree (ADT) classifier construction, handles only data measurement errors present in the numerical attributes of the training data sets.

C. Suggestions for future work

Special techniques or ideas or better plans are needed to find and correct data errors other than data measurement errors that are present in the numerical attributes of the training data sets. Special pruning techniques may help us to improve the performance and accuracy of the decision tree classifier construction.

REFERENCES

[1] Jiawei Han, Micheline Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann, second edition, 2006. pp. 285–292
[2] Introduction to Machine Learning Ethem Alpaydin PHI MIT Press, second edition. pp. 185–188
[3] SMITH Tsang, Ben Kao, Kevin Y. Yip, Wai-Shing Ho, and Sau Dan Lee “Decision Trees for Uncertain Data” IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING,VOL.23, No.1, JANUARY 2011
[4] Hsiao-Wei Hu, Yen-Liang Chen, and Kwei Tang “A Dynamic Discretization Approach for Constructing Decision Trees with a Continuous Label” IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL.21, No.11, NOVEMBER 2009
[5] R.E. Walpole and R.H. Myers, Probability and Statistics for Engineers and Scientists. Macmillan Publishing Company, 1993.
[6] T.M. Mitchell, Machine Learning. McGraw-Hill, 1997.
[7] A. Asuncion and D. Newman, UCI Machine Learning Repository, http://www.ics.uci.edu/mlearn/MLRepository.html, 2007.
[8] U.M. Fayyad and K.B. Irani, “On the Handling of Continuous –Valued Attributes in Decision tree Generation”, Machine Learning, vol. 8, pp. 87-102, 1996.
[9] J.R. Quinlan, “Induction of Decision Trees,” Machine Learning, vol. 1, no. 1, pp. 81-106, 1986.