Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 6, June 2014, pg.176 – 183

RESEARCH ARTICLE

Improved Accuracy for Decision Tree Algorithm Based on Unsupervised Discretization

Ihsan A. Kareem [1], **Mehdi G. Duaimi** [2]

[1,2]Dept. of Computer Science, College of Science, University of Baghdad, Baghdad, Iraq
[1] ihsana.kareem@uokufa.edu.iq; [2] mehdi_duaimi@scbaghdad.edu.iq

Abstract - A decision tree is an important classification technique in data mining classification. Decision trees have proved to be valuable tools for the classification, description, and generalization of data. Work on building decision trees for data sets exists in multiple disciplines such as signal processing, pattern recognition, decision theory, statistics, machine learning and artificial neural networks. This paper deals with the problem of finding the parameter settings of decision tree algorithm in order to build an accurate tree.

The proposed technique is an unsupervised filter. The suggested discretization applies on C4.5 algorithm to construct a decision tree. The improvement on C4.5 algorithm includes two phases: the first phase is discretization all continuous attributes instead of dealing with numerical values. The second phase is constructing a decision tree model and evaluates performance. It has been experimented on three data sets. All those data files are picked up from the popular UCI the (University of California at Irvine) data repository. The results obtained from experiments show C4.5 after discretization better than C4.5 before discretization.

Keywords- Decision tree classification, C4.5 algorithm, discretization, continuous attributes

1. Introduction

Data mining is the process of discovering meaningful new correlations, patterns and trends by sifting through large amounts of data stored in repositories, using pattern recognition technologies as well as statistical and mathematical techniques [1]. Classification is a form of data analysis that extracts model describing important data classes. Such models, called classifiers, predict categorical (discrete, unordered) class labels. For example, a classification model can be built to categorize bank loan applications as either safe or risky [2]. Decision tree induction is the learning of decision trees from class labeled training tuples. A decision tree is a flowchart like tree structure, where each internal node (non leaf node) denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (or terminal node) holds a class label. The topmost node in a tree is the root node [3].

The C4.5 algorithm is an extension to ID3 developed by Quinlan Ross. It is also based on Hunt's algorithm. C4.5 handles both categorical and continuous attributes to build a decision tree. In order to handle continuous attributes, C4.5 splits the attribute values into two partitions based on the selected threshold such that all the values above the threshold as one child and the remaining as another child. It also handles missing attribute values. C4.5 uses gain ratio as an attribute selection measure to build a decision tree [4].

An optimization problem of machine learning algorithms consists of finding the parameter settings in order to build an accurate model for a given domain. The main problem is dealing with continuous attributes. For example if there are N different values of attribute A in the set of instances D, there are N - 1 thresholds that can be used for a test on attribute A. Every threshold gives unique subsets D1 and D2; hence, the value of the splitting criterion is a function of the threshold.

2. Related Works

For surveying the problem of improving decision tree classification algorithm for large data sets, several algorithms have been developed for building DTs of large data sets.

Kohavi& John 1995 [5], searched for parameter settings of C4.5 decision trees that would result in optimal performance on a particular data set. The optimization objective was 'optimal performance' of the tree, i.e., the accuracy measured using 10-fold cross-validation. Liu X.H 1998 [6], proposed a new optimized algorithm of decision trees. On the basis of ID3, this algorithm considered attribute selection in two levels of the decision tree and the classification accuracy of the improved algorithm had been proved higher than ID3. Liu Yuxun&XieNiuniu 2010 [7], solving the problem of a decision tree algorithm based on attribute importance is proposed. The improved algorithm uses attribute-importance to increase the information gain of attributes which has fewer attributions and compares ID3 with improved ID3 by an example. The experimental analysis of the data shows that the improved ID3 algorithm can get more reasonable and more effective rules. Gaurav & Hitesh 2013 [8], propose C4.5 algorithm which is improved by the use of L'Hospital Rule, this simplifies the calculation process and improves the efficiency of decision making algorithms. The aim is to implement the algorithms in a space effective manner and response time for the application will be promoted as the performance measures. The system aims to implement these algorithms and graphically compare the complexities and efficiencies of these algorithms.

3. Decision Tree Construction Algorithm

The algorithm constructs a decision tree starting from a training set TS, which is a set of cases, or tuples in the database terminology. Each case specifies values for a collection of attributes and for a class. Each attribute may have either discrete or continuous values. Moreover, the special value unknown is allowed to denote unspecified values. The class may have only discrete values [9].

3.1 Divide and Conquer

The skeleton of Hunt's method for constructing a decision tree from a set T of training cases is elegantly simple. Let the classes be denoted as: {C1, C2,... , Ck}. There are three possibilities:

1. T contains one or more cases: all belonging to a single class C. The decision tree for T is a leaf identifying class C.
2. T contains no cases: The decision tree is again a leaf, but the class to be associated with the leaf must be determined from information other than T.
3. T contains cases that belong to a mixture of classes: In this situation, the idea is to refine T into subsets of cases that are, or seem to be heading towards, single-class collections of cases.

3.2 Possible Tests Considered

Most classifier building systems define a form for possible tests and then examine all tests of this form. Conventionally, a test involves just one attribute, because this makes the tree easier to understand and sidesteps the combinatorial explosion that results if multiple attributes can appear in a single test [10].

C4.5 contains mechanisms for proposing three types of tests:

1. The "standard" test on a discrete attribute, with one outcome and branch for each possible value of that attribute.
2. A more complex test, based on a discrete attribute, in which the possible values are allocated to a variable number of groups with one outcome for each group rather than each value.
3. If attribute A has continuous numeric values, a binary test with outcomes A <= Z and A > Z, based on comparing the value of A against a threshold value Z.

4. Information Gain and Gain Ratio

The ingormation gain is a measure in the information theory, which characterizes the impurity of an arbitrary collection of examples. If the target attribute takes on *c* different values, then the entropy S relative to this n-wise classification is defined as:

$$Entropy\ (s) = \sum_{i=1}^{n} -Pi\ \log_2 Pi \quad \dots\ Equation\ (1)$$

Where pi is the proportion/probability of S belonging to class *i*. Logarithm is base 2 because entropy is a measure of the expected encoding length measured in bits [11].

The information gain measures the expected reduction in entropy by partitioning the examples according to this attribute. The information gain, Gain(S, A) of an attribute A, relative to the collection of examples S, is defined as:

$$Gain\ (S, A) = Entropy\ (S) - \sum_{Values\ (A)} \frac{|Sv|}{|S|}\ \dots\ Equation\ (2)$$

Where*Values (A)*is the set of all possible values for attribute A, and *Sv* is the subset of S for which the attribute A has value *v*. We can use this measure to rank attributes and build the decision tree where at each node is located the attribute with the highest informationgain among the attributes not yet considered in the path from the root [11].

In order to reduce the effect of the bias resulting from the use of information gain, a variant known as **Gain Ratio** was introduced by the Australian academic Ross Quinlan in his influential system C4.5. Gain Ratio adjusts the information gain for each attribute to allow for the breadth and uniformity of the attribute values [12]. The default splitting criteria used by c4.5 is gain ratio, an information based measure that takes into account different numbers (and different probabilities) of tests outcomes [10].

C4.5, a successor of ID3, uses an extension to information gain known as gain ratio, which attempts to overcome this bias. It applies a kind of normalization to information gain using a “split information” value defined analogously with Info (D) as:

$$SplitInfo\ A(D) = -\sum_{j=1}^{v} \frac{|Dj|}{|D|} \times \log_2\left(\frac{|Dj|}{|D|}\right) \dots\ Equation\ (3)$$

This value represents the potential information generated by splitting the training data set, D, into v partitions, corresponding to the v outcomes of a test on attribute A. Note that, for each outcome, it considers the number of tuples having that outcome with respect to the total number of tuples in D which is ($|D|$). It differs from information gain, which measures the information with respect to classification that is acquired based on the same partitioning. The gain ratio is defined as:

$$GainRatio\ (A) = \frac{Gain\ (A)}{SplitInfo\ A\ (D)}\ \dots\ Equation\ (4)$$

The attribute with the maximum gain ratio is selected as the splitting attribute. Note, however, that as the split information approaches 0, the ratio becomes unstable [13].

5. Discrediting Numeric Attributes

Some classification and clustering algorithms deal with nominal attributes only and cannot handle ones measured on a numeric scale. To use them on general datasets, numeric attributes must first be “discredited” into a small number of distinct ranges. Although most decision tree and decision rule learners can handle numeric attributes, some implementations work much more slowly when numeric attributes are present because they repeatedly sort the attribute values [14].

Unsupervised discretization algorithms are the simplest to use and implement. They only require the user to specify the number of intervals and/or how many data points should be included in any given interval [4].

The goal of discretization is to reduce the number of values a continuous attribute assumes by grouping them into a number, n, of intervals (bins). Tables (1), (2), (3) presents discretization process for bank marketing, diabetes and credit approval data sets respectively.

Table (1) Discretization Process for bank data

Attributes	Range	Code
Age	23 - 30.6	A
	30.6 - 38.2	B
	38.2 - 45.8	C
	45.8 - 53.4	D
	53.4 – 61	E
Balance	2827 - 11082.6	A
	11082.6 - 24992.2	B
	24992.2 - 38901.8	C
	38901.8 - 52811.4	D
	52811.4 - 66721	E
Day	4 - 9.4	A
	9.4 - 14.8	B
	14.8 - 20.2	C
	20.2 - 25.6	D
	25.6 – 31	E
Duration	5 - 688.4	A
	688.4 - 1371.8	B
	1371.8 - 2055.2	C
	2055.2 - 2738.6	D
	2738.6 – 3422	E

Table (2) Discretization Process for diabetes data

Attributes	Range	Code
Perg	0 - 8.5	A
	8.5 - 17	B
Plas	0 - 99.5	A
	99.5 - 199	B
Pres	0 - 61	A
	61 - 122	B
Skin	0 - 49.5	A
	94.5 - 99	B
Insu	0 - 423	A
	423 - 846	B
Mass	0 - 33.55	A
	33.55 - 67.1	B

Pedi	0.078 - 1.249	A
	1.249 - 2.42	B
Age	21 - 51	A
	51 - 81	B

Table (3) Discretization Process for credit approval data

Attributes	Range	Code
A2	13.75 - 35.91	A
	35.91 - 58.08	B
	58.08 - 80.25	C
A3	0 - 9.33	A
	9.33 - 18.66	B
	18.66 - 28	C
A8	0 - 9.5	A
	9.5 - 19	B
	19 - 28.5	C
A11	0 - 22.33	A
	22.33 - 44.66	B
	44.66 - 67	C
A14	0 - 666.66	A
	666.66 - 1333.33	B
	1333.33 - 2000	C
A15	0 - 33333.33	A
	33333.33 - 66666.66	B
	66666.66 - 100000	C

6. Experimental Results

This section contains the experimental results of the proposed approach, briefing on Visual Basic.NET (VB.NET) as a programming language used in the implementation. The evaluation results obtained from the testing process is presented. In general, the performance of a classifier is mainly measured by the classification accuracy that is how often the future prediction for a record with unknown class label. It has experimented on three data sets. All these data files are picked up from the popular UCI the (University of California at Irvine) data repository. Table (4) shows the details of those data files.

Table (4) Data Sets Properties

Data Set Name	Number of Instances	Number of Attributes	Attribute Types	Missing Values
Bank Marketing	4521	13	Continuous & Nominal	No
Diabetes	768	9	Continuous	No
Credit Approval	690	16	Continuous & Nominal	No

The results are obtained using VB.NET on three data sets by two-fold cross-validation to test the accuracy of C4.5 and improved C4.5 classifiers.

Table (5) shows the model accuracy and error rate in percent for two classifiers (C4.5 and improved C4.5) to build a model for training data. Figures (1), (2) show a comparison of classifier accuracy and error rate.In other words, each figure shows the difference in accuracy and error rate for all data sets between the two classification algorithms (C4.5, improved C4.5).

Table (5) Model Accuracy and Error Rate for All Data Sets

Data Set Name	Accuracy (%)		Error Rate (%)	
	C4.5	IC4.5	C4.5	IC4.5
Bank Marketing	93.98363	94.89051	6.01636	5.10948
Diabetes	70.96354	71.875	29.03645	28.125
Credit Approval	78.98550	82.02898	21.01449	17.97101

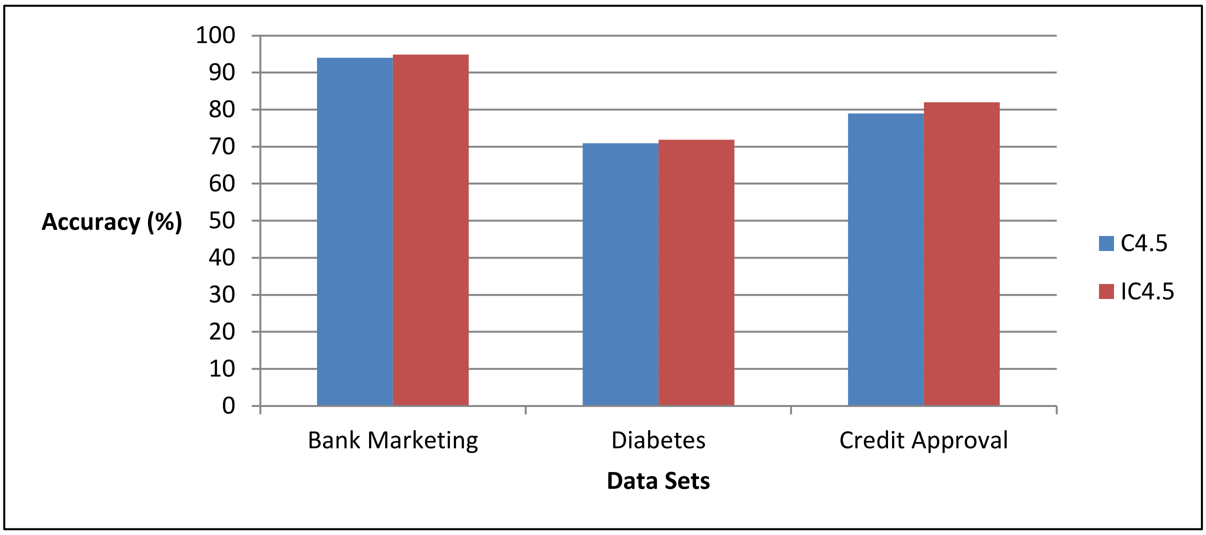

Fig1.Comparison of Classifiers Accuracy

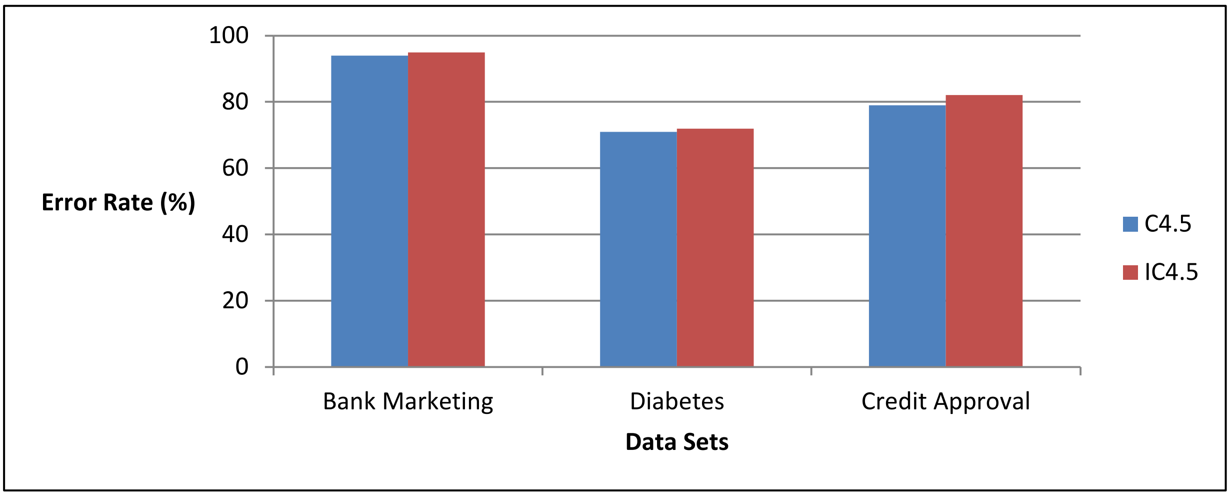

Fig2. Comparison of Classifiers Error Rate

7. Conclusions and Future works

This paper focuses on C4.5 algorithm. The C4.5classifier is one of the most classic classification algorithms on data mining and most often used, decision tree classifiers. C4.5 algorithm performs well in constructing decision trees and extracting rules from the data set. The objective of this work has been to demonstrate that the technology for building decision trees from examples is fairly robust. The purpose of this paper is to increase the classification accuracy when building a classification model. Cross-validation technique is adopted in order to evaluate the performance of the C4.5 and IC4.5 algorithms. The improvement is proposed to implement C4.5 algorithm for construction a decision tree model; some useful conclusions are listed in the following points:

1. Dealing with continuous attributes may lead to inappropriate classification.
2. The experimental results reveal the accuracy and size of the tree for bank data set; where the accuracy is increased by 1 %.For diabetes data set; the accuracy is increased by 1%. For credit approval data set; the accuracy is increased by 4%.

Also, the results of this paper point to several interesting directions for future work:

1. Currently, the proposed approach doesn't handle the missing values. So, applying our proposed approach to the domains that involve missing values is the main work in the future.
2. An approach for improvement the accuracy could be extended to other parameters such as tree size or execution time.

References

[1] Daniel T. Larose, "*Discovering Knowledge in DataAn Introduction to Data Mining*",John Wiley & Sons, 2005.

[2] Mehmed Kantardzic, "*Data Mining: Concepts, Models, Methods, and Algorithms*", ISBN:0471228524, John Wiley & Sons, 2003.

[3] Sushmtta, Mitra, &Tinku, Acharya, "*Data Mining Multimedia, Soft Computing, and Bioinformatics*", John Wiley & Sons, Inc, 2003.

[4] Krzysztof J. Cios, WitoldPedrycz, Roman W. Swiniarski, & Lukasz A.Kurgan, "*Data Mining A Knowledge Discovery Approach*", Springer Science Business Media, LLC, 2007.

[5] Tea Tusar, "*Optimizing Accuracy and Size of Decision Trees*", Department of Intelligent Systems, Jozef Stefan Institute, Jamova 39, SI-1000 Ljubljana, Slovenia, 2007.

[6]Weiguo Yi, Jing Duan, &Mingyu Lu, "*Optimization of Decision Tree Based on Variable Precision Rough Set*", International Conference on Artificial Intelligence and Computational Intelligence, 2011.

[7] Liu Yuxun, &XieNiuniu, "*Improved ID3 Algorithm*", IEEE, 2010.

[8] Gaurav L. Agrawal, & Prof. Hitesh Gupta, "*Optimization of C4.5 Decision Tree Algorithm for Data Mining Application*", International Journal of Emerging Technology and Advanced Engineering, Volume 3, Issue 3, March 2013.

[9] Salvatore Ruggieri, "*Efficient C4.5*", IEEE Transactions on Knowledge and Data Engineering, Vol 14, No. 2, pp. 438-444, 2002.

[10] J. R . Quinlan, "*Improved Use of Continuous Attributes in C4.5*", Journals of Artificial Intelligent Research, 1996.

[11] Anand Bahety ,"*Extension and Evaluation of ID3 – Decision Tree Algorithm*",Department of Computer Science University of Maryland, College Park.

[12] Max Bramer, "*Principles of Data Mining*", Springer-Verlag London Limited, 2007.

[13]Jiawei Han, MichelineKamber, & Jian Pei, "*Data Mining Concepts and Techniques*", Third Edition, Morgan Kaufmann, 2012.

[14] Ian H. Witten, Eibe Frank, & Mark A. Hall, "*Data Mining Practical Machine Learning Tools and Techniques*", Third Edition, Morgan Kaufmann, 2011.