ISSN:0975-9646

Improved Decision Tree Induction Algorithm with Feature Selection, Cross Validation, Model Complexity and Reduced Error Pruning

A. S. Galathiya[#1], A. P. Ganatra[#2] and C. K. Bhensdadia[#1]

[#1]*Faculty of Technology, D. D. University – Nadiad, India*
[#2]*Charotar Institute of Technology- Changa, India*

***Abstract*— Data mining is the process of finding new patterns. Classification is the technique of generalizing known structure to apply to new data. Classification using a decision tree is performed by routing from the root node until arriving at a leaf node. To model classification process, decision tree is used. Decision can handle both continuous and categorical data.**

In this research work, Comparison is made between ID3, C4.5 and C5.0. Among these classifiers C5.0 gives more accurate and efficient output with comparatively high speed. Memory usage to store the ruleset in case of the C5.0 classifier is less as it generates smaller decision tree. This research work supports high accuracy, good speed and low memory usage as proposed system is using C5.0 as the base classifier. The classification process here has low memory usage compare to other techniques because it generates fewer rules. Accuracy is high as error rate is low on unseen cases. And it is fast due to generating pruned trees.

This research work proposed C5.0 classifier that performs feature selection, cross validation, reduced error pruning and model complexity for original C5.0 in order to reduce the optimization of error ratio.

In this paper, feature selection, cross validation, reduced error pruning and model complexity are the techniques which are described as those are used in the proposed system.

Feature selection is used for dimensionality reduction. It reduces the attribute space of a feature set. It is to remove irrelevant data attributes. One way to get a more reliable estimate of predictive is by cross- validation. By increasing the model complexity, accuracy of the classification is increases. By applying reduced error pruning technique, overfitting problem of the decision tree is solved.

Using this proposed system; Accuracy will be gained about 1 to 3 %. The classification error rate is reduced compare to the existing system and within less time the decision tree is constructed.

***Keywords*— REP, Decision Tree induction, C5 classifier, KNN, SVM**

I INTRODUCTION

This paper describes first the comparison of best-known supervised techniques in relative detail. Then it produces a critical review of comparison between supervised algorithms like Decision Tree with Naive Bayes, KNN, SVM, Neural Networks and Bayesian Classifier. It is not to find that which classification learning algorithm is superior to others, but under which conditions a particular method can significantly outperform others on a given application problem.

In data mining, Decision tree structures are a common way to organize classification schemes. Classification using a decision tree is performed by routing from the root node until arriving at a leaf node. The research work is made up from ID3, C4.5 and C5 classifier. In many applications, rulesets are preferred because they are simpler and easier to understand than decision trees. [4] Both C4.5 and C5.0 can produce classifiers expressed either as decision trees or rulesets, but C4.5's ruleset methods are slow and high memory is required. C5.0 embodies new algorithms for generating rulesets with improved features.

This research work supports high accuracy, good speed and low memory usage. Memory usage is low compare to other classifier because it generates fewer rules. Accuracy is high as error rate is low on unseen cases. And it is fast due to generating pruned trees.

II SURVEY ON CLASSIFICATION ALGORITHMS

a. Decision trees

Decision trees are trees that classify instances by sorting them based on feature values. Each node in a decision tree represents a feature in an instance to be classified, and each branch represents a value that the node can assume. Instances are classified starting at the root node and sorted based on their feature values [7]. Decision tree rules provide model transparency so that a user can understand the basis of the model's predictions, and therefore, be comfortable acting on them and explaining them to others.

b. Naive Bayes Algorithm

The Naive Bayes algorithm (NB) can be used for both binary and multiclass classification problems. Naive Bayes algorithm builds and scores models extremely rapidly; it scales linearly in the number of predictors and rows.

Naive Bayes algorithm makes predictions using Bayes' Theorem which derives the probability of a prediction from the underlying evidence. Bayes' Theorem states that the probability of event A occurring given that event B has occurred (P(A|B)) is proportional to the probability of event B occurring given that event A has occurred multiplied by the probability of event A occurring ((P(B|A)P(A)).

c. K-nearest neighbour

Nearest neighbour classifier is based on learning by analogy. The training samples are described by n dimensional numeric attributes. When given an unknown sample, a k-nearest neighbour classifier searches the pattern space for the k training samples that are closest to the unknown sample. "Closeness" is defined in terms of Euclidean distance, where the

Euclidean distance, where the Euclidean distance between two points, X=(x1,x2,……,xn) and Y=(y1,y2,….,yn) is

$$d(X, Y)=\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

The unknown sample is assigned the most common class among its k nearest neighbours. When k=1, the unknown sample is assigned the class of the training sample that is closest to it in pattern space.

d. SVM

The second method we can use for training purposes is known as Support Vector Machine (SVM) classification. SVM is a type of machine learning algorithm derived from statistical learning theory. A property of SVM classification is the ability to learn from a very small sample set.

e. Neural Networks

A neural network, when used for classification, is typically a collection of neuron-like processing units with weighted connections between the units. Back propagation is a neural network learning algorithm. Neural network learning is also referred to as connectionist learning due to the connections between units.

f. Comparison between Classification Algorithms

i) Advantages:

TABLE I ADVANTAGES OF SUPERVISED ALGORITHM

Decision Tree	Naive Bayes	K- Nearest Neighbor	SVM	Neural Networks
Easily Observed & develop generated rules	Fast, highly scalable model building (parallelized) and scoring.	Robust to noisy training data and effective if the training data is large.	More accurate than Decision Tree classification.	high tolerance of noisy data and ability to classify patterns for untrained data

ii) Features Comparison:

TABLE II FEATURE COMPARISON BETWEEN DIFFERENT CLASSIFICATION ALGORITHMS

Feature	Decision Tree	Naive Bayes	K- Nearest Neighbour	Support Vector Machine	Neural Networks
Learning Type	Eager Learner	Eager Learner	Lazy learner	Eager Learner	Eager Learner
Speed	Fast	Very fast	Slow	Fast with active learning	Slow
Accuracy	Good in many domains	Good in many domains	High – Robust	Significantly high	Good in many domains
Scalability	Efficient for small data set	Efficient for large data set	-	-	Low
Interpretability	Good	-	-	-	Bad
Transparency	Rules	No rules (black box)	Rules	No rules (black box)	black box
Missing val Interpretation	Missing Value	Missing Value	Missing Value	Sparse data	-

iii) Comparison based on Classification parameter:

Here rating is given 4 is the best one and 1 is the worst [9].

TABLE III COMPARISON BASED ON CLASSIFICATION PARAMETER

Parameters	Decision Tree	Naive Bayes	K- Nearest Neighbour	Support Vector Machine	Neural Networks
Accuracy in general	2	1	2	4	3
Speed of learning with respect to number of attributes and the number of instances	3	4	4	1	1
Speed of classification	4	4	1	4	4
Tolerance to missing values	3	4	1	2	1
Tolerance to irrelevant Attributes	3	2	2	4	1
Tolerance to redundant Attributes	2	1	2	3	2
Dealing with discrete /binary /continuous attributes	4	3 (not continuous)	3 (not discrete)	2(not discrete)	3(not discrete)
Tolerance to noise	2	3	1	2	2
Dealing with Over fitting	2	3	3	2	1
Attempts for incremental Learning	2	4	4	2	3
Explanation ability/transparency of knowledge/classifications	4	4	2	1	1

III C5.0 CLASSIFIER

The classifier is trained and tested first. Then the resulting decision tree or rule set is used to classify unseen data. C4.5 is the successor algorithm of ID3. C4.5 is the successor algorithm of C5. C5.0 algorithm has many features like:

- C5.0 algorithm can respond on noise and missing data.
- C5.0 provides boosting.
- A large decision tree may be difficult to read and comprehend.
- C5.0 provides the option of viewing the large decision tree as a set of rules which is easy to understand.
- Overfitting is solved by the C5.0 and ***Reduce error pruning technique***.
- C5.0 can also predict which attributes are relevant in classification and

which are not. This technique, known as ***Winnowing*** is especially useful while dealing with high dimensional datasets.

a. Algorithm C5

Input: Example, Target Attribute, Attribute
Output: decision tree

Algorithm:

- Check for the base class
- Construct a DT using training data
- Find the attribute with the highest info gain (A_Best)
- For each ti ϵ D, apply the DT to determine its class Since the application of a given tuple to a DT is relatively straightforward.

base cases are the following for the algorithms C4.5 and C5.0:

• All the examples from the training set belong to the same class (a tree leaf labeled with that class is returned).

• The training set is empty (returns a tree leaf called failure).

• The attribute list is empty (returns a leaf labeled with the most frequent class or the disjunction of all the classes).

OUTPUT: *decision tree which classifies the data correctly*

b. Comparison- Current Algorithms

i) C4.5 - Improvements from ID3 algorithm

- Handling both continuous and discrete attributes - In order to handle continuous attributes, C4.5 creates a threshold and then splits the list into those whose attribute value is above the threshold and those that are less than or equal to it.
- Handling training data with missing attribute values - C4.5 allows attribute values to be marked as '?' for missing. Missing attribute values are simply not used in gain and entropy calculations.
- Handling attributes with differing costs.
- Pruning trees after creation - C4.5 goes back through the tree once it's been created and attempts to remove branches that do not help by replacing them with leaf nodes.

ii) C5.0 - Improvements from C4.5 algorithm [32]

- **Speed** - C5.0 is significantly faster than C4.5 (several orders of magnitude)
- **Memory usage** - C5.0 is more memory efficient than C4.5 C5.0 commonly uses an order of magnitude less memory than C4.5 during ruleset construction.
- ***Accuracy:*** The C5.0 rulesets have noticeably lower error rates on unseen cases. Sometimes the C4.5 and C5.0 rulesets have the same predictive accuracy, but the C5.0 ruleset is smaller.
- **Smaller decision trees** - C5.0 gets similar results to C4.5 with considerably smaller decision trees.
- **Support for boosting** - Boosting improves the trees and gives them more accuracy.
- **Weighting** - C5.0 allows you to weight different attributes and misclassification types.

c. Problem of Current System

Issues in data mining with decision trees

There are some issues in learning decision trees which include:

- Determining how deeply to grow the decision tree
- Handling continuous attributes
- Choosing an appropriate attribute selection measure
- Handling training data with missing attribute values
- Handing attributes with differing costs
- Improve computational efficiency

d. Solution via proposed system:

- **Avoiding over-fitting the data**

In case of noisy data or in case of too small training set, is really difficult to classify.

In either of these cases, this simple algorithm can produce trees that *over-fit* the training examples.

There are several approaches to avoiding over-fitting in decision tree learning. These can be grouped into two classes:

i. Approaches that stop growing the tree earlier, before it reaches the point where it perfectly classifies the training data,
ii. Approaches that allow the tree to over-fit the data and then post prune the tree.

First approach is more direct compare to post pruning. Post pruning is the acceptable approach as in case of the first approach it is difficult to know that when to stop. It is still not known that how to determine the correct tree size.

- **Incorporating Continuous-Valued Attributes**

The ID3 is restricted to attributes that take on a discrete set of values.

In the solution of the above problem, continuous-valued decision attributes can be incorporated into the learned tree. This can be able to do by dynamically defining new discrete-valued attributes that partition the continuous attribute value into a discrete set of intervals.

IV. RESEARCH WORK

a. Input Parameter:

Example

Examples are input data which is expected to correctly classify.

Attributes

Input to algorithm consists of a collection of training cases, each having a tuple of values for a fixed set of attributes (or independent variables) A = {A1, A2,...., A_k} and a class attribute (or dependent variable).

Target attributes

An attribute A_a is described as continuous or discrete, if it is the C4.5 algorithm. In case of the ID3, Attributes are of type only numerical or nominal. The class attribute (target attributes) C is discrete and has values C1, C2,...,C_x.

OUTPUT: *decision tree which classifies the data correctly*

b. Proposed Algorithm:

Feature Selection

Feature selection is known as variable selection, feature reduction, attribute selection or variable subset selection. Feature Selection is used dimensionality reduction technique in machine learning and data mining. The application where Feature Selection is used, are text classification and web mining. Feature Selection builds the faster model by reducing the number of features, and also helps remove irrelevant, redundant and noisy features.

Reduced Error Pruning

Reduced Error Pruning is a technique in machine learning that reduces the size of decision trees by removing sections of the tree that provide little power to classify instances. The dual goal of pruning is reduced complexity of the final classifier as well as better predictive accuracy by the reduction of overfitting and removal of sections of a classifier that may be based on noisy or erroneous data.

Algorithm:

- Create a root node for the tree
- Check for the base case
- **Apply Feature Selection using Genetic Search**
- **bestTree = Construct a DT using training data**
- **Perform Cross validation**
  1. **Divide all examples into N disjoint subsets, E = E_1, E_2, ..., E_N**
  2. **For each i = 1, ..., N do**
     - **Test set = E_i**
     - **Training set = E - E_i**
     - **Compute decision tree using Training set**
     - **Determine performance accuracy P_i using Test set**
  3. **Compute N-fold cross-validation estimate of performance = (P_1 + P_2 + ... + P_N)/N**
- **Perform Reduced Error Pruning technique**
- Find the attribute with the highest info gain (A_Best)

  Perform Model complexity
- Partition S into S1,S2,S3... according to the value of A_Best
- Repeat the steps for S1, S2, S3
- Classification :
- For each ti ϵ D, apply the DT to determine its class

Cross Validation

Cross-Validation is the method of evaluating and comparing learning algorithms by dividing data into two segments: one used to learn or train a model and the other used to validate the model.

Model Complexity

By increasing the complexity of the model, classification accuracy is increased. Complexity of model is increased by changing parameters.

V. EXPERIMENTAL RESULTS

After using functionality of Global Pruning and Cross validation with C5 classifier, the results are found like:

Dataset	w/o Global Pruning and Cross Validation		with Global Pruning and Cross Validation	
	Training Data Error	Testing Data Error	Mean Error	Std. Error
Soybean.data	5%	15.5%	11%	1.7%
Banding.data	0.8%	6%	3%	0.9%

By applying two functionalities, the result found is improved naturally.

VI. CONCLUSION AND FUTURE WORK

The important task of classification process is to classify new and unseen sample correctly. C5.0 is a classifier which gives efficient classification in less time compare to other classifier. Memory usage is less in generating decision tree. The main objective of research is related to improve accuracy and generate small decision tree. For this proposed system is developed on the bases of C5.0 algorithm. In the new system, Feature selection, Cross validation, reduced error pruning and model complexity are the facility provided in C5.0 algorithm. The accuracy is gained by 1-3% by the new system.

Thus as the further scope, The implementation is done for the new features like: Feature Selection, Reduced Error Pruning, Cross Validation and Model Complexity. By implementing the diversities of algorithm using RGUI with weka packages, the classification accuracy is improved.

References

[1] Sohag Sundar Nanda, Soumya Mishra, Sanghamitra Mohanty, Oriya Language Text Mining Using C5.0 Algorithm, (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 2 (1) , 2011

[2] cTomM.Mitchel, McGrawHil, Decision Tree Learning, Lecture slides for textbook Machine Learning, , 197

[3] Zuleyka Díaz Martínez, José Fernández Menéndez, Mª Jesús Segovia Vargas See5 Algorithm versus Discriminant Analysis, Spain.

[4] XindongWu · Vipin Kumar · J. Ross Quinlan · Joydeep Ghosh · Qiang Yang ·Hiroshi Motoda · Geoffrey J. McLachlan · Angus Ng · Bing Liu · Philip S. Yu ·Zhi-Hua Zhou · Michael Steinbach · David J. Hand · Dan Steinberg Top 10 algorithms in data mining © Springer-Verlag London Limited 2007

[5] J.R, QUINLAN , Induction of Decision Trees, New South Wales Institute of Technology, Sydney 2007, Australia

[6] Rulequest Research, "Data Mining Tools See5 and C5.0, http:/lwww.rulequest.com/see5-info.html, 1997-2004

[7] Decision Trees for Business Intelligence and Data Mining: Using SAS Enterprise Miner "Decision Trees— What Are They?"

[8] Thair Nu Phyu, "Survey of Classification Techniques in Data Mining", International MultiConference of Engineers and Computer Scientists 2009 Vol I IMECS 2009, March 18 - 20, 2009, Hong Kong

[9] S. B. Kotsiantis, Department of Computer Science and Technology "Supervised Machine Learning: A Review of Classification Techniques" - , University of Peloponnese, Greece

[10] Classification: basic Concepts, Desision Tree, and model evalution

[11] Terri Oda , Data Mining Project, April 14, 2008

[12] Matthew N. Anyanwu manyanwu, Sajjan G. Shiva sshiva, Comparative Analysis of Serial Decision Tree Classification Algorithms
[13] Maria Simi , Decision tree learning
[14] Osmar R. Zaïane, 1999, Introduction to Data Mining, University of Alberta
[15] J. R. B. COCKETT, J. A. HERRERA, Decision tree reduction, University of Tennessee, Knoxville, Tennessee
[16] Hendrik Blockeel , Jan Struyf, Efficient Algorithms for Decision Tree Cross-validation, Department of Computer Science, Katholieke Universiteit Leuven, Belgium
[17] S. Rasoul Safavian and David Landgrebe, A Survey of Decision Tree Classifier Methodology, School of Electrical Engineering Purdue University, West Lafayette
[18] Floriana Esposito, Donato Malerba, and Giovanni Semeraro, A Comparative Analysis of Methods for Pruning Decision Trees, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 19, NO. 5, MAY 1997
[19] Paul E. Utgoff, Neil C. Berkman, Jeffery A. Clouse, Decision Tree Induction Based on Efficient Tree Restructuring, Department of Computer Science, University of Massachusetts, Amherst, MA 01003
[20] Niks, Nikson , Decision Trees, Introduction to machine learning,
[21] Ron Kohavi Ross Quinlan , Decision Tree Discovery, Blue Martini Software 2600 Campus Dr. Suite 175, San Mateo, CA & Samuels Building, G08 University of New South Wales, Sydney 2052 Australia
[22] Paul E. Utgoff , Incremental Induction of Decision Trees, Department of Computer and Information Science, University of Massachusetts, Amherst, MA 01003
[23] Matti K¨a¨ari¨ainen , Tuomo Malinen, Tapio Elomaa, Selective Rademacher Penalization and Reduced Error Pruning of Decision Trees, Department of Computer Science, University of Helsinki, Institute of Software Systems, Tampere University of Technology, Tampere, Finland
[24] Michael Kearns, Yishay Mansour, A Fast, Bottom-Up Decision Tree Pruning Algorithm with Near-Optimal Generalization, AT&T Labs, Tel Aviv University
[25] Zijian Zheng, Constructing new attributes for decision tree learning, Basser Department of Computer Science, the university of Sydney, Australia
[26] Tan, Steinbach, Kumar , Data Mining Classification: Basic Concepts, Decision Trees, and Model Evaluation
[27] Emily Thomas, DATA MINING: DEFINITIONS AND DECISION TREE EXAMPLES, Director of Planning and Institutional Research, State University of New York
[28] Kurt Hornik, The RWeka Package August 20, 2006
[29] Kurt Hornik, Christian Buchta, Achim Zeileis, Open-Source Machine Learning: R Meets Weka, WU Wirtschaftsuniversit at Wien
[30] Zhengping Ma, Eli Lilly and Company, Data mining in SAS® with open source software, SAS Global Forum 2011
[31] Simon Urbanek, Package 'rJava', Jan 2, 2012
[32] M. Govindarajan, Text Mining Technique for Data Mining Application, World Academy of Science, Engineering and Technology 35 2007

Authors

Avni S. Galathiya is a student of Master of Technology in Computer Engineering at Dharmsinh Desai University, Nadiad, Gujarat, India. She is also an Assistant Professor at R. C. Technical Institute of Computer department, Ahmedabad, Gujarat, India. She has received her B.E. Computer Engineering degree from Dharmsinh Desai University, Nadiad, Gujarat, India in 2007. She has joined M. Tech. at Dharmsinh Desai University, Nadiad, Gujarat, India in 2010. Her current research interest includes Data Mining with classification (C5.0 classifier).

Amit P. Ganatra (B.E.-'00-M.E. '04-Ph.D.* '11) has received his B.Tech. and M.Tech. degrees in 2000 and 2004 respectively from Dept. of Computer Engineering, DDIT-Nadiad from Gujarat University and Dharmsinh Desai University, Gujarat and he is pursuing Ph.D. in Information Fusion Techniques in Data Mining from KSV University, Gandhinagar, Gujarat, India and working closely with Dr.Y.P.Kosta (Guide). He is a member of IEEE and CSI.
His general research includes Data Warehousing, Data Mining and Business Intelligence, Artificial Intelligence and Soft Computing. In these areas, he is having good research record and published and contributed over 70 papers (Author and Co-author) published in referred journals and presented in various international conferences. He has guided more than 90 industry projects at under graduate level and 47 dissertations at Post Graduate level.
He is concurrently holding Associate Professor (Jan 2010 till date), Headship in computer Engineering Department (since 2001 to till date) at CSPIT, CHARUSAT and Deanship in Faculty of Technology-CHARUSAT (since Jan 2011 to till date), Gujarat. He is a member of Board of Studies (BOS), Faculty Board and Academic Council for CHARUSAT and member of BOS for Gujarat Technological University (GTU). He was the founder head of CE and IT departments of CITC (now CSPIT).

C.K. Bhensdadia is Professor and Head of Department of Computer Engineering at the Dharmsinh Desai University, Nadiad- Gujarat, India. He received a B.E. degree from Dharmsinh Desai University, Nadiad -Gujarat, India in 1990, and M.Tech. degree from IIT Mumbai in 1996. He is currently pursuing Ph.D. His main areas of research are the Genetic Algorithms and Data Mining. He has presented numerous papers on these topics in International journals & Conferences. He is also principal investigator for many research projects, sponsored by Govt. of India.