Extracting Rules from Artificial Neural Networks utilizing TREPAN

Maimuna Rangwala and Gary R. Weckman
Industrial and Manufacturing Systems Engineering
Ohio University
Athens, Ohio, 45701

Abstract

Artificial Neural Networks (ANN) have been applied to a wide variety of real world classification problems. However they act like black boxes and lack explanation capability. It is essential to understand the reasoning behind such a learning system in domain specific applications. Representing the extracted knowledge in the form of if-then rules is an excellent method of explaining the output of an ANN model. Decision trees can be easily represented in the form of if-then rules hence, extracting decision trees is one of the best methods of explaining an ANN. The TREPAN algorithm mimics the behavior of an ANN in the form of decision trees by approximating the function represented by the network. This paper involves evaluating results obtained from TREPAN against the benchmark C4.5 algorithm to test for accuracy, computational speed and comprehensibility for various problem domains. The classification accuracy is assessed using a confusion matrix and kappa statistic.

Keywords: Artificial Neural Networks, Decision Trees, TREPAN, C4.5

1. Introduction

Artificial Neural Networks(ANN) offer a means of efficiently modeling large and complex problems in which there may be hundreds of independent variables that have many interactions, classification being one of them. ANNs generate their own implicit rules by learning from examples. Since their resurgence in the 1980's ANNs have been applied to a variety to problem domains [1] such as, speech recognition [2], speech generation [3], medical diagnostics [4], game playing [5] and robotics [6]. The generalization ability of ANN has proved to be on par or superior to other learning systems over a wide range of applications [7]. One of the most pronounced drawbacks of ANN is their lack of explanation capability. They act like 'black boxes' and do not give any reasoning behind the conclusion of a learning system [8]. However it is essential to understand the basis of decisions of this type of computer support systems for various reasons. Although ANNs are known to be robust classifiers, they have found limited use in decision critical applications such as medical systems. The availability of a system that would provide an explanation of the input/output mappings of an ANN in the form of rules would thus be very useful. Rule extraction is one such system that tries to reveal to the user, how the ANN arrived at its decision in the form of if-then rules. A decision tree is a special type of graph drawn in the form of a tree structure. It consists of internal nodes, each associated with a logical test and its possible consequences and can be easily expressed in the form of if-then rules.

2. Literature Review

Decision trees classify data through recursive partitioning of the data set into mutually exclusive subsets which best explain the variation in the dependent variable under observation[9][10]. Decision trees classify instances (data points) by sorting them down the tree from the root node to some leaf node. This leaf node gives the classification of the instance. Schmitz et al [11] has proposed an algorithm ANN-DT that extracts decision trees from trained ANN. Craven and Shavlik [12] have developed the TREPAN algorithm that mimics the behavior of an ANN in the form of decision trees. This paper uses the TREPAN algorithm to extract knowledge from the ANN.

2.1 TREPAN

The TREPAN algorithm [12] [13] developed by Craven, is a novel rule-extraction algorithm that mimics the behavior of an ANN. Given a trained Neural Network, TREPAN extracts decision trees that provide a close approximation to the function represented by the network. In this thesis we are concerned with its application to trained ANN models, although it can be applied not only to ANN but also to a wide variety of learned models. TREPAN uses a concept of recursive partitioning similar to other decision tree induction algorithms. In contrast to the depth-first growth used by other decision tree algorithms, TREPAN expands using the best first principle. The node which increases the fidelity of the tree when expanded is deemed the best.

In conventional decision tree induction algorithms the amount of training data decreases as one traverses down the tree by selecting splitting tests. Thus there is not enough data at the bottom of the tree to determine class labels accurately. In contrast TREPAN uses an ‘Oracle’ to create additional data points in addition to the training samples by querying the ANN. Since the ANN represents the underlying phenomena of the data, the network itself is used as the ‘Oracle’ to create more data points when necessary to determine a split in the decision tree. This additional data allows the decision tree algorithm to learn from larger samples and can avoid problems associated with the lack of examples. These additional data points improve the accuracy for the splitting tests at lower levels of a decision tree which is typically a problem with conventional decision tree learning algorithms. This ‘Oracle’ ensures that there is a minimum sample of instances (data points) available at a node by creating them before determining a splitting test. If the number of instances at the node is less than the minimum sample (user specified), then TREPAN will query the ANN and create a number of data points till the number of samples is equal to the minimum sample size for a given decision node. Trepan uses the m-of-n test to partition the part of the instance space (dataset) covered by a particular branch node. An m-of-n expression (a Boolean expression) is satisfied when at least m features out of n conditions hold true. For example, consider four features a, b, c and d; if m=3 and n=4, then if any 3 of the features of the given set of 4 are satisfied then the test will pass and the tree will continue through that node.

2.2 C4.5

The C4.5 [14] algorithm developed by Quinlan is one of the most widely used decision tree learning algorithms. It is a non-incremental algorithm, which means that it derives its classes from an initial set of training instances. The classes derived from these instances are expected to work for all future test instances. The algorithm uses the greedy search approach i.e. selects the best attribute and never looks back to reconsider earlier choices. The C4.5 algorithm searches through the attributes of the training instances and finds that attribute that best separates the data. If this attribute perfectly classifies the training set then algorithm stops else it recursively works on the remaining m subsets (m = the remaining possible values of the attribute) to get their best attribute. A fundamental part of any algorithm that constructs a decision tree from a dataset is the method in which it selects attributes at each node of the tree for splitting so that the depth of the tree is minimal. C4.5 uses the concept of Gain Ratio for this purpose.

Gain measures how well a given attribute separates training examples into its target classes. The attribute with the highest information is selected. Information gain calculates the *reduction in entropy* (or *gain in information*) that would result from splitting the data into subsets based on an attribute. The information gain of example set S on attribute A is defined as,

$$Gain(S,A) = Entropy(S) - \sum \frac{|S_v|}{|S|} Entropy(S_v)$$

where: $|S_v|$ is a subset of instances of *S*, *A* takes the value *v*, and *S* is the number of instances

Entropy is a measure of the amount of information in an attribute. The higher the entropy, the more is the information required to completely describe the data. Hence, when building the decision tree, the idea is to decrease the entropy of the dataset until a subset is reached that is pure (a leaf), has zero entropy and represents instances that all belong to one class. Entropy is given by,

$$Entropy(S) = -\sum p(I) \log_2 p(I)$$ where: $p(I)$ is the proportion of *S* belonging to class I

3. Methodology

The flowchart in Figure 1 outlines the steps involved in this analysis. A real world dataset is selected for evaluation. After initial pre-processing of data for errors, outliers, etc., a neural network model was built experimenting with different kinds of network layouts and transfer functions. The best model obtained is then fed to the decision tree induction algorithm TREPAN for classification. The dataset is also analyzed using a baseline decision tree induction algorithm, C4.5. Results obtained by TREPAN are then compared to C4.5 to test for accuracy, computational speed and comprehensibility. In addition to tree size, two parameters: classification accuracy and the kappa statistic are used to assess each of the classifier’s performance.

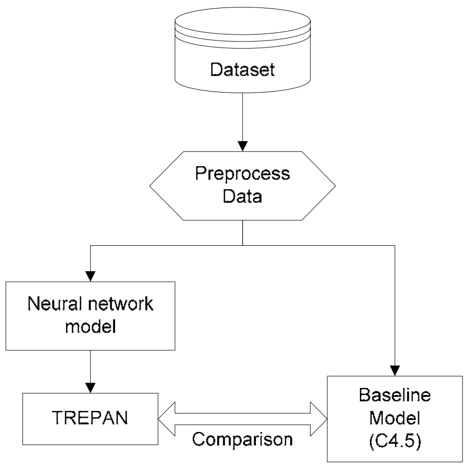

Figure 1: Methodology

3.1 Dataset and variables

Many researchers have worked on the problem of modeling the CO_2 corrosion process due to its significance in predicting the corrosion rate. The corrosion dataset is used in this paper. The corrosion dataset is a collection of characteristic composition of 15 Venezuelan crude oils used to predict the ability of a crude oil to offer corrosion inhibition in a CO_2 environment [15]. In addition to process complexity, the small size of the dataset (98 instances) makes it even more challenging to model through traditional statistical means. Attributes such as API, sulphur, total nitrogen, TAN(total acid number), saturates, aromatics, resins, asphaltenes, vanadium, nickel and percentage of crude oil are used for the analysis. The TAN or total acid number indicates the amount of oxidation that has taken place in a fluid. A detailed description of the test conditions used to determine these oil characteristics are outlined by Hernandez [15]. These attributes are used to predict the inhibiting capacity and hence the corrosion rate given by the formula,

$$Inhibiting\ capacity = 1 - \frac{Corrosion\ rate_{crude\ oil}}{Corrosion\ rate_{blank}}$$

A sample of the dataset and the class sizes for the regression output are shown in
Table 1 and
Table 2 respectively.

Table 1: Corrosion-sample dataset

Sr. No.	API	S (%)	Total N	TAN	Sat.	Arom.	resins	Asph.	V	Ni	%Crude oil	% Inhibition
1	33.3	0.566	923	0.02	62.2	34.8	3	0	5	5	20	0.814013
2	20.7	1.1	3276	0.11	43.2	32.6	17.6	6.6	125	25	80	0.992309
3	20.4	2.42	3450	0.74	30	45.1	16.4	8.6	362	44	50	0.891
4	8.5	3.7	6948	3.35	13.3	47.8	28.8	10.1	458	95	1	0.812019
5	11	2.6	4340	4.78	21.1	50.1	19.7	9.2	409	52	20	0.986

Table 2: Corrosion Class labels

%Inhibition	Class 1	Class 2	Class 3	Class 4	Class 5
	0.75-0.849	0.85-0.889	0.89-0.949	0.95-0.979	0.98 and above

3.2 ANN modeling

NeuroSolutions [16] software was used for neural network modeling. The data was split as 70%, 15% and 15% for training, cross-validation and testing respectively. An 11-5-3-1 MLP network with a hyperbolic tangent function was used to train the corrosion model for 20000 epochs. Figure 2 shows the graph of the actual output against the ANN output with an r (correlation co-efficient) value of 0.942.

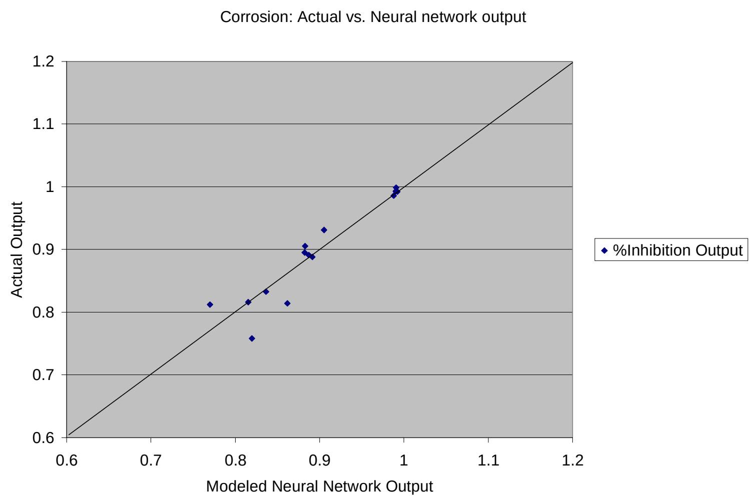

Figure 2: Actual vs. Model output

3.3 Performance measures

Classification accuracy and kappa statistic was used to measure the predictive performance of the classifiers. The classification accuracy or error rate is the percentage of correct predictions made by the model over a data set. It is assessed using the confusion matrix. A confusion matrix is a matrix plot of predicted versus actual classes with all the correct classifications depicted along the diagonal of the matrix. It presents the number of correctly classified instances, incorrectly classified instances and overall classification accuracy. Consider a two class problem as shown in Table 3. The four possible outcomes in this case are the true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN). As the name suggests a false positive is when a negative instance is incorrectly classified as a positive, and a false negative is when a positive instance is incorrectly classified as a negative.

Table 3: Confusion Matrix-example

Class		**Predicted Class**	
		Yes	**No**
Actual Class	**Yes**	TP	FN
	No	FP	TN

The accuracy of the classifier is given by the formula,

$$Accuracy(\%) = \frac{(TP + TN)}{(TP + FN + FP + TN)} \times 100$$

A confusion matrix is a primary tool in visualizing the performance of a classifier. However, it does not take into account the fact that some misclassifications are worse than others. To overcome this problem we use a measure called the Kappa Statistic [17] which considers the fact that correct values in a confusion matrix are due to chance agreement. The Kappa statistic is defined as,

$$\hat{k} = \frac{P(A) - P(E)}{1 - P(E)}$$

where: $P(A)$ is the proportion of times the model values were equal to the actual value and,
$P(E)$ is the expected proportion by chance.

4. Results

TREPAN generated a model with a classification accuracy of 85.71%, a Kappa Statistic of 0.440 and a decision tree having 12 nodes and 13 leaves. The decision tree generated by TREPAN is depicted in Figure 3. C4.5 gave a low model accuracy of 57.14%, A Kappa Statistic of 0.808 and a decision tree having 11 nodes and 12 leaves. The confusion matrixes for both models are shown in Table 4.

Table 4: Corrosion Confusion Matrix (TREPAN versus C4.5)

(TREPAN)						**(C4.5)**				
Actual/Desired	**Cl 1**	**Cl 2**	**Cl 3**	**Cl 4**	**Cl5**	**Cl 1**	**Cl 2**	**Cl 3**	**Cl 4**	**Cl 5**
Class 1	**4**	0	0	0	0	**3**	2	0	0	0
Class 2	1	**1**	0	0	0	0	**0**	1	0	0
Class 3	0	0	**4**	0	0	0	1	**2**	0	1
Class 4	0	0	0	**0**	1	0	0	0	**0**	0
Class 5	0	0	0	0	**3**	0	0	0	1	**3**
Classification Accuracy (%)	**80.00%**	**100.00%**	**100.00%**	**0.00%**	**75.00%**	**100.00%**	**0.00%**	**66.67%**	**0.00%**	**75.00%**
Total Accuracy (%)	**85.71%**					**57.14%**				

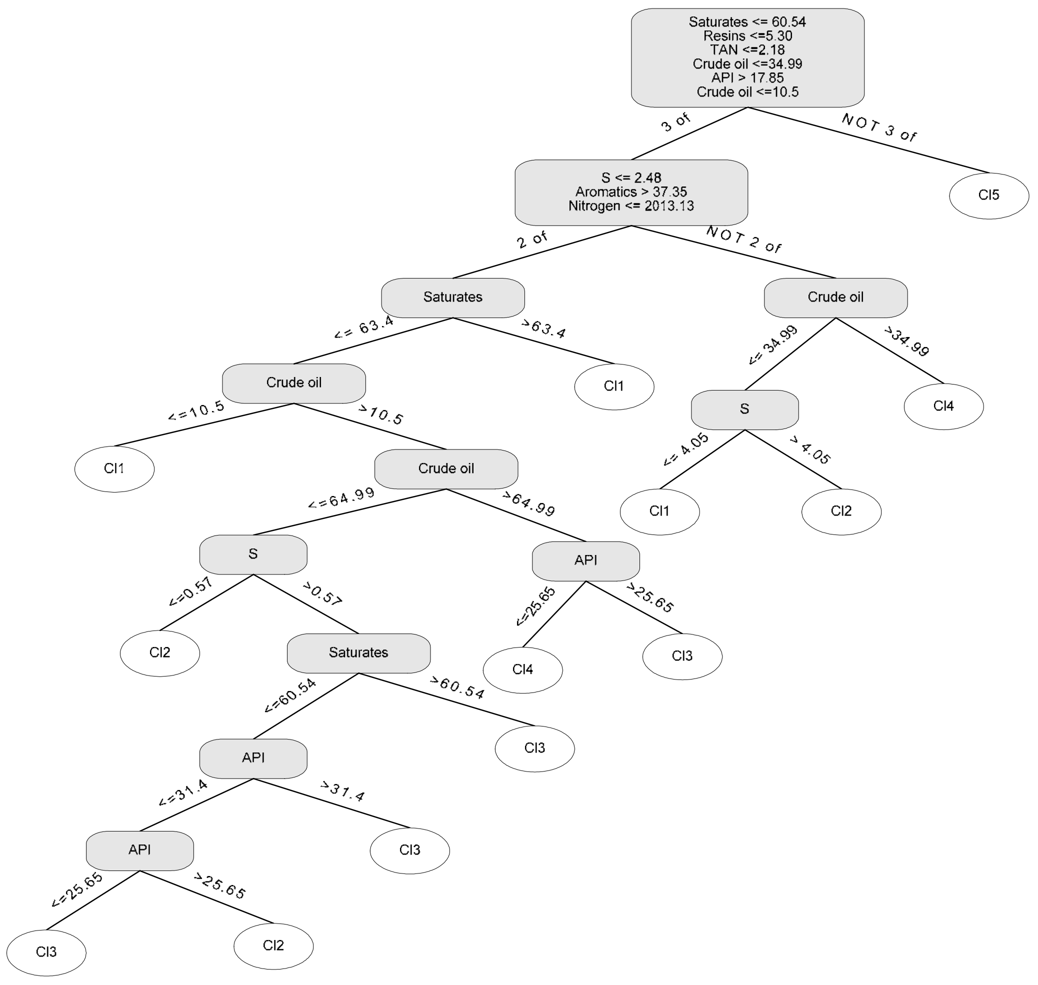

Figure 3: TREPAN Decision Tree

The results indicate that the use of a ANN to create more datasets improves the ability to create a more accurate decision tree for estimating the inhibition of crude oils. The Kappa statistic indicates that the TREPAN model, in addition to being more accurate, tends to place misclassifications closer to the diagonal (perfect classifications) as shown in Table 4.

5. Conclusions

Based on the research performed on a few diverse databases, the TREPAN algorithm significantly outperforms the C4.5 algorithm in terms of classification accuracy. TREPAN has demonstrated its ability to be excellent instrument in combining the power of the highly incomprehensible neural network black box and the straightforwardness of decision trees. The performance of TREPAN versus C4.5 was measured in both the classification accuracy the kappa statistic measurement. The Kappa value for TREPAN was nearly twice as much as that of C4.5. The generality of TREPAN stems from the fact that its use is not limited to neural networks alone. This research has shown that TREPAN is a better tool at decision tree induction than the most commonly used algorithm, C4.5.

REFERENCES:

[1]. A. B. Tickle A. B., Andrews R., Golea M., Diederich J. (1998). The truth will come to light: Directions and challenges in extracting the knowledge embedded within trained artificial neural networks, *IEEE Trans. Neural Networks*, Nov., 9 (6): 1057–1068.

[2]. Waibel A. H., (1989) Modular construction of time-delay neural networks for speech recognition, *Neural computation*, 1: 39 – 46.

[3]. Sejnowski T. J., Rosenberg C. R., (1986) A parallel network that learns to read aloud. Technical report JHU/EECS-86/01, John Hopkins Univerity.

[4]. Jabri M., Pickard S., Leong P., (1992) ANN based classification for heart defibrillators. In Moody J. E., Hanson S. J.,Lippmann R., P., (Eds.), *Advances in neural information processing systems*, pp 637-644, San Mateo, CA.

[5]. Tesauro G. J., (1992) Practical issues in temporal difference learning, *Machine Learning*, 8.

[6]. Hsu G. T., Simmons R., (1991) Learning football evaluation for a walking robot. In proceedings of eighth international workshop on Machine learning, Jun, pp 303-307, Evanston, IL.

[7]. Shavlik J. W., Mooney R. J., Towell G. G. (1991) Symbolic and Neural Net Learning Algorithms: An Empirical Comparison, *Machine Learning*, 6: 111–143.

[8]. Towell G. G., Shavlik J. W. (1993). Extracting Refined Rules from Knowledge-Based Neural Networks, *Machine Learning*, 13: 71–101.

[9]. Biggs, D., de Ville, B. and Suen, E. (1991) A method of choosing multiway partitions for classification and decision tree. *Journal of Applied Statistics* 18(1): 49-62.

[10]. Liepins, G., Goeltz, R. and Rush, R. (1990) Machine learning techniques for natural resource data analysis. *AI Applications* 4(3): 9-18.

[11]. Schmitz, G. P.J., Aldrich, C., & Gouws, F. S. (1999). ANN-DT: An algorithm for extraction of decision trees from artificial Neural Networks. IEEE Transactions on Neural Networks, 10(6), 1392-1401.

[12]. Craven, M. W., (1996) Extracting Comprehensible models from trained Neural Networks, PhD Thesis, Computer Science Department, University of Wisconsin, Madison, WI.

[13]. Craven, M. W., and Shavlik, J. W., (1996) Extracting tree-structured representations of trained networks. In Advances in Neural Information Processing Systems, volume 8, pages 24–30, Denver, CO, MIT Press.

[14]. Quinlan, J. R. (1993). C4.5: Programs in machine learning. San Mateo, CA: Morgan Kaufmann.

[15]. Hernandez, S., Nesic, S., Weckman, G., Ghai, V., (2005). Use of artificial neural networks for predicting crude oil effect on CO2 corrosion of carbon steels.

[16]. NeuroSolutions, (1995). Software developed and distributed by Neurodimension Incorporated [http://www.neurosolutions.com/products/ns/].

[17]. Cohen, J. (1960). A coefficient of agreement for nominal scales. Educational and Psychological Measurement, 20, 37-4