A Comparative Study On Neural Network Based Rule Extraction Algorithm And C4.5

MOHAMED A.El-SHARKAWY MOUSTAFA M. SYIAM* SHAYMAA M.KHATER**
Information System Dept Computer Science Dept Information System Dept
Ain Shams University
Faculty Of Computer And Information Sciences,Cairo,Egypt

Abstract: Classification is one of the data mining problems receiving great attention recently in the database community. This paper proposes a comparative study to highlight the significant difference between the C4.5 decision tree based algorithm and the neural network approach for classification and rule extraction. We compare the rules generated by the C4.5 with that generated by the RX (Neural Network based Algorithm). This is experimentally evaluated in different domains. The Experimental results demonstrate that rules extracted from neural networks are comparable with those of decision trees in terms of predictive accuracy, number of rules and average number of conditions for a rule.

Key-Words: Data mining, classification, rule extraction, decision tree, neural network, machine learning.

1 Introduction

One of the data mining problems is the classification. Various classification algorithms are designed to tackle the problem by researchers in different fields such as machine learning. Most algorithms are basically based on decision trees [1]. On the other hand, the use of neural networks in classification is not uncommon in machine learning [2]. There has been a significant amount of work devoted to the development of algorithms that extract rules from neural networks [4, 5, 6, 7, 9]. Recent papers can be found in [12, 13]

In this paper, we describe the most popular classification decision tree algorithm, the C4.5 algorithm [3], and how rules can be extracted from these decision trees. Then, we describe an algorithm for rule extraction from neural networks, the RX algorithm [4]. This algorithm consists of three major phases:

1. Network construction and training: This phase constructs and trains a three layer neural network based on the number of attributes and the number of classes and the chosen dataset.
2. Network pruning: The pruning phase aims at removing the redundant links and units without increasing the classification error rate of the network.
3. Rule extraction: This phase extracts the classification rules from the pruned network.

Due to space limitation we omit the discussion of the first two phases. Details of these phases can be found in [10, 11]. In section 2.1, we describe the C4.5 decision tree based algorithm. In section 2.2, we describe the RX algorithm. In section 3, case studies on Iris-plants, Breast- cancer, Mushroom classification, voting-records databases are presented to show the results of applying the two algorithms for rule extraction process. Finally, a brief conclusion is given in section 4.

2 Background

2.1 C4.5 decision-tree algorithm

This algorithm was proposed by Quinlan (1993) [3]. It generates a classification decision tree for the given data-set by recursive partitioning of data. The algorithm considers all the possible tests that can split the data set and selects a test that gives the best information gain. A major concept involved in this algorithm is the Entropy (also called the information). The entropy concept is used to find the most significant parameter in characterizing the classifier.

Calculation of Entropy consists of three phases. Assuming that:

n: number of classes,

t: number of training examples,
t (nj) : number of training examples satisfying the class (nj),
d: number of attributes on which we split the instances,

- Entropy before branching: for the full data set is calculated as
  E = Sum {-(t (nj)/t) * [log (t (nj) / t / log (n)]}.
- Then Entropy of each branch: for each value di of the attribute d is calculated
  Ei = Sum{-[t(di(nj))/ t(di)] *[log (t(di(nj))/ t(di)) / log(n)]}.
- Then Summation to entropy of branching:
  e = sum {t (di)/t) * ei}.
- Entropy gain is then calculated as the difference between original full entropy and the branching entropy. This is the measure of how significant the parameter is in characterizing the classifier.
  G = E-e.
  The algorithm chooses the next split so as to maximize the gain or minimize the entropy. Rules in the form of " if A and B then C" are then generated where A and B are the rule antecedents while C is the rule consequence. Every path from the root to the leaf is converted to an initial rule by regarding all the test conditions appearing in the path as the rule antecedents while regarding the class label held by the leaf as the rule consequence. After that, each initial rule is generalized by removing antecedents that do not seem helpful for distinguishing a specific class from other classes. After all initial rules are generalized, they are grouped into rule sets corresponding to the classes respectively.

2.2 A Rule Extraction Algorithm

The rule extraction algorithm, RX, is a neural network based approach. It consists mainly of four steps [4] given below:

1) Apply a clustering algorithm to find clusters of hidden node activation values.
2) Enumerate the discretized activation values and compute the network outputs. Generate rules the network output in terms of the hidden unit activation values.
3) For each hidden unit, enumerate the input values that lead to them and generate a set of rules to describe the hidden units discretized values in terms of the inputs.
4) Merge the two sets of rules obtained in the previous two steps to obtain rules that relate the inputs and outputs.

To cluster the activation values, we used a clustering algorithm which consists of the following steps:

1) Let ε Є (0,1). Let D be the number of discrete activation values in the hidden unit. Let δ_1 be the activation values in the hidden unit for the first pattern in training set. Let H(1) = δ_1, count(1) = 1, and sum (1) = δ_1, set D = 1.
2) For each pattern p_i , i = 2, 3, .. k in the training set:
   - Let δ be its activation value.
   - If there exists an index ĵ such that
     ı δ – H(ĵ)ı = min ı δ – H(J) and
     ı δ – H(ĵ)ı ≤ ε where j Є {1,2,….D}
3) Replace H by the average of all activation values that have been clustered into this cluster:
   H (j):= sum (j)/count (j), j = 1, 2,,,, D.

3 Experimental Results

In this section, we describe four datasets and show how the two algorithms are applied to extracting rules. Summary of the results on all the datasets are then given.

3.1 Iris-Plants Database

This database was obtained from the University Of California-Irvine. The dataset contains 3 classes, each one with 50 examples of flower: Iris-Setosa, Iris-Versicolor and Iris Verginica. One class is linearly separable from the other 2, the latter are not linearly separable from each other. There are 4 attributes involved: Sepal-length, Sepal-width, Petal-length and Petal-width. Each one of these attributes takes a continuous value.

On running the RX Algorithm on the dataset

We trained a multilayer perceptron neural network on 120 exemplars, 40 from each class. The other 30 exemplars were used for testing. The best neural network was obtained by using 4 input units, 4 hidden units and 3 output units. we used the sigmoid activation function where

$$F(x_i , w_i) = 1/1+ \exp (-x_i^{lin}) \qquad (1)$$

Where x_i^{lin} =ß x_i is the scaled and offset activity. The accuracy of the neural networks is summarized in Table 5. This network is shown in Fig. 1.

After clustering the hidden activation values, the results of clustering algorithm were as shown in Table 1.

Table 1
Results of Clustering The Hidden Activation Values of The Original Network for Iris-Plants Datasets.

Hidden node	discrete values	Values	Epsilon (ε)
1	2	0.982675, 0.997066	0.006
2	2	0.980079, 0.991431	0.006
3	3	0.353401,0.625751, 0.933157	0.15
4	2	0.5897, 0.990897	0.006

These values produced a total of 24 possible outputs for the network. The accuracy of the network with these discrete activation values is summarized in Table 6. The accuracy was nearly the same as that achieved by the original network.

The following rules relating the input to the output were generated:

- If 2 < Petal Length ≤ 4.7 and Petal Width < 1.7 then Iris-Versicolor.
- Else If Petal Length ≤ 2 then Iris-Setosa.
- Default Rule : Iris-verginica

On running the C4.5 on the same data set the following rules were generated:

- If Petal Length < = 2 then Iris-Setosa .
- Else If 2 < Petal Length < 4.5 then Iris-Versicolor
- Else If Petal Width > 1.6 then Iris – Verginica.
- Default Rule: Iris-Verginica.

We observed that on running the two algorithms on Continuous data, the accuracy of the rules generated by the two algorithms were nearly the same (98%). C4.5 generated 5 rules while RX algorithm generated 4 rules. The average number of conditions in rules generated by RX was 2 conditions per rule. C4.5 generated less number of conditions per rule. It was also observed that the accuracy rates of the RX rules were slightly greater than those of the network with discrete hidden-unit activation values mentioned in Table 6.

3.2 Breast-Cancer Database

The database for the Wisconsin Breast-Cancer diagnoses is available from the University of California-Irvine repository [8]. The dataset consists of 699 examples, of which 458 examples are classified as Benign, and 241 as malignant. Nine attributes were involved. Each attribute takes an ordinal integer from 1 to 10 (10 values).

On running the RX Algorithm on the dataset

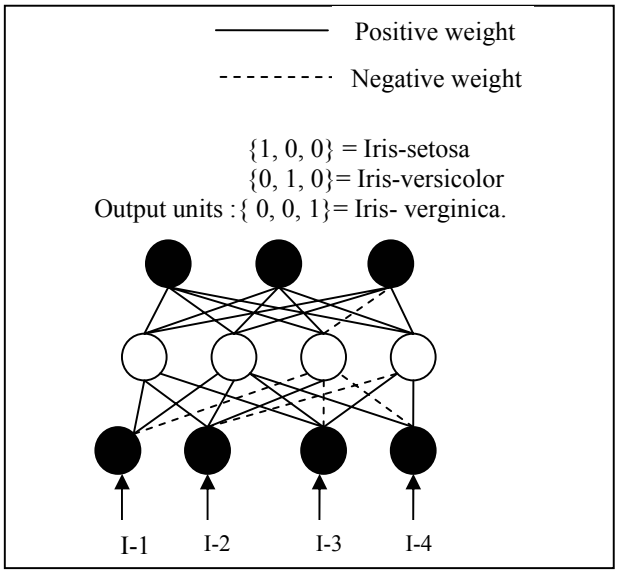

Fig.1: The original network for the Iris-data problem. The inputs are numbered sequentially from 1 to 4.

We trained a multilayer perceptron neural network on 609 exemplars, 200 representing the first class and 409 for the second class. The rest of exemplars were used for testing. The best neural network was obtained by using 9 input units, 5 hidden units and 2 output units. we used the same activation function as (1).The accuracy of the neural networks is summarized in Table 5.

After clustering the hidden activation values, the results of clustering algorithm were as shown in Table 2.

Table 2
Results of Clustering The Hidden Activation Values of The Original Network for Breast-Cancer Datasets.

Hidden node	discrete values	Values	Epsilon (ε)
1	2	0.063614, 0.995775	0.8
2	2	0.002085, 0.926747	0.5
3	2	0.047176, 0.957903	0.6
4	2	0.215124, 0.998552	0.5
5	2	0.009801, 0.916333	0.3

These values produced a total of 32 possible outputs for the network. The accuracy of the network with these discrete activation values is summarized in Table 6. The accuracy was nearly the same as that achieved by the original network.

The following rules relating the input to the output were generated:

- If Clump Thickness ≤ 6 and uniformity of cell shape ≤ 2 and Bland Chromatin ≥ 5 and Normal Nucleoli ≤ 5 and uniformity of cell size ≤ 7 then Benign.
- Else If Clump Thickness ≤ 6 and uniformity of cell size ≤ 7 and Bare Nucleoli ≤ 3 and Bland Chromatin ≥5 and Normal Nucleoli ≤ 5 then Benign.
- Default Rule: Malignant.

On running the C4.5 on the same data set the following rules were generated:

- If uniformity of cell size < 5 and Bare Nucleoli ≤ 3 then Benign.
- Else If uniformity of cell shape < 5 and Bare Nucleoli ≤ 3 and Bland Chromatin ≥5 then Benign.
- Default Rule: Malignant.

From the rules generated, we observed that on running the two algorithms on discrete data, the number of rules generated were nearly the same (3 rules). C4.5 generated more rules with less number of conditions than those generated by RX algorithm (3 conditions in C4.5 versus 4 conditions in RX). However, the accuracy rates of the rules generated by RX were nearly the same as those of the network with discrete hidden-unit activation values.

3.3 Mushroom Database

The database is available from the University of California-Irvine repository [8]. The dataset includes descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota Family (pp. 500-525). The dataset consists of 8124 examples, of which 4208 examples were classified as definitely edible, and 3916 as definitely poisonous. There are twenty two attributes involved, all are nominally valued.

On running the RX Algorithm on the dataset

We trained a multilayer perceptron neural network on 3000 exemplars, 1500 representing the first class and 1500 for the second class. We used 500 exemplars for testing. The best neural network was obtained by using 23 input units, 5 hidden units and 2 output units. we used the same function as .

The accuracy of the neural networks is summarized in Table 5.

After clustering the hidden activation values, the results of clustering algorithm were as shown in Table 3.

Table 3
Results of Clustering The Hidden Activation Values of The Original Network for Mushroom Classification Datasets.

Hidden node	discrete values	Values	Epsilon (ε)
1	2	0.283213, 0.96962	0.6
2	2	0.302175, 0.962967	0.55
3	2	0.561401, 0.884518	0.2
4	2	0.070543, 0.692342	0.29
5	2	0.184726, 0.493508	0.23

These values produced a total of 32 possible outputs for the network. The accuracy of the network with these discrete activation values is summarized in Table 6. The accuracy was nearly the same as that achieved by the original network.

The following rules relating the input to the output were generated:

- If odor = {creosote, fishy, pungent, spicy, foul} and cap-color = {buff, pink, white, yellow, gray, brown} and then poisonous.
- Else If odor = none and stalk-surface- above- ring = silky and spore-print-color = {green, white} and cap-color = {buff, pink, white, yellow} and ring type = pendant then edible.
- Default Rule: edible.

On running the C4.5 on the same data set the following rules were generated:

- If odor = {creosote, fishy, pungent, spicy, foul} then poisonous.
- Else If odor = none and ring-type = pendant then poisonous.
- Else If odor = none and spore-print-color = {green, white} then poisonous.
- Default Rule: edible

3.4 Voting-Records Database

The database is .available from the University of California Irvine repository [8]. The dataset includes votes for each of the U.S. House of representatives Congressmen on the 16 key votes identified by the CQA. It contains two classes: democrat and republican. The dataset consists of 435 examples, of which 267 were classified as democrats and 168 as republican.16 attributes were involved, all are Boolean valued.

On running the RX Algorithm on the dataset

We trained multilayer perceptron neural network on 350 exemplars, 220 representing the first class and

130 for the second class. We also used 100 exemplars for testing. The best neural network was obtained by using 16 input units, 4 hidden units and 2 output units. The accuracy of the neural networks is summarized in Table 5.

After clustering the hidden activation values, the results of clustering algorithm were as shown in Table 4.

Table 4
Results of Clustering The Hidden Activation Values of The Original Network for Voting-Records Datasets.

Hidden node	discrete values	Values	Epsilon (ε)
1	2	0.209507, 0.760213	0.3
2	2	0.098366, 0.976561	0.6
3	3	0.089107, 0.475764, 0.786149	0.21
4	2	0.225655, 0.697355	0.28

These values produced a total of 24 possible outputs for the network. The accuracy of the network with these discrete activation values is summarized in Table 6.

The following rules relating the input to the output were generated:

- If physician-fee-freeze = n and El-Salvador-aid = n or y then Democrat**.**
- If physician-fee-freeze = y El-Salvador-aid= y, adoption-of-the-budget-resolution= y, mx-missile = y and aid-to-Nicaraguan-contras = y then Democrat.
- Else if physician-fee-freeze = y El-Salvador-aid= y, adoption-of-the-budget-resolution= y, duty-free-exports= y, mx-missile = y, aid-to-Nicaraguan-contras = n**,** water-project-cost-sharing = n, immigration = n, handicapped-infants = y then Republican.
- Default rule: Democrat

On running the C4.5 on the same data set the following rules were generated:

- If physician-fee-freeze = n and El-Salvador-aid = n or y then Democrat**.**
- Else if physician-fee-freeze = y, El-Salvador-aid = y, adoption-of-the-budget-resolution= y, duty-free-exports = y and mx-missile = y then Democrat.
- Else if physician-fee-freeze = y, El-Salvador-aid=y, adoption-of-the-budget-resolution = y and superfund-right-to-sue = y then republican.
- Else if physician-fee-freeze = y, El-Salvador-aid= n and adoption-of-the-budget-resolution= y then republican..
- Default rule: Democrat.

Table 5
Accuracy of the original network

	Accuracy (%)	
	Training data	Testing data
Iris dataset	93.3%	93.3%
Breast-cancer dataset	99.60%	94%
Mushroom dataset	99%	98.60%
Voting-records dataset	100%	91%

Table 6
Accuracy of the network *after* clustering the activation values of the hidden units

	Accuracy (%)	
	Training data	Testing data
Iris dataset	91.6%	91.1%
Breast-cancer dataset	95.50%	94%
Mushroom dataset	97%	98.40%
Voting-records dataset	98%	91%

From the rules generated, we observed that on running the two algorithms on discrete data with large number of inputs, the two algorithms still have nearly the same accuracy, but different number of rules generated and different number of conditions per rule. The accuracy rates of the rules generated by the RX algorithm were the same as those of the network after clustering the hidden-unit activation values.

Fig. 2, Fig. 3 and Fig. 4 show the accuracy, number of rules and the average number of conditions in the rules generated by the two approaches. We can see that the two approaches have nearly the same accuracy. The number of rules generated by the neural network approach is less than that generated by the C4.5. However the average number of conditions in rules generated by the neural networks approach is greater than that of C4.5.

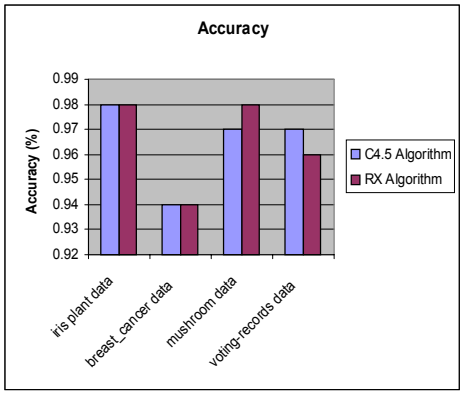

Fig.2: Accuracy of the rule extracted from the Neural Networks and C4.5 rule for the four datasets.

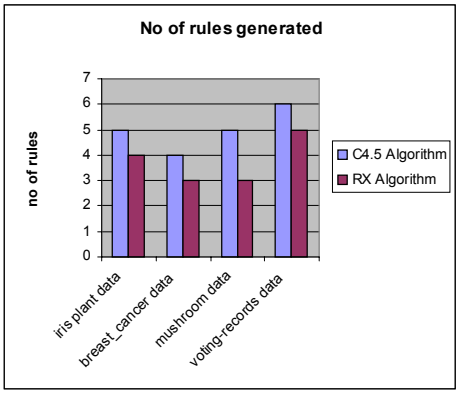

Fig.3: Number of rules extracted from the Neural Networks and C4.5 rule for the four datasets.

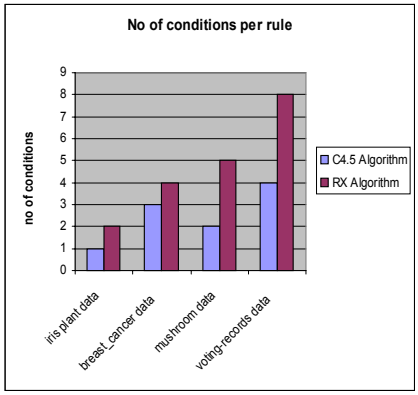

Fig.4: Average number of conditions per rule for the four datasets.

4 Conclusion

In this paper we presented a comparative study on different datasets to highlight the significant difference between the decision tree based algorithm, the C4.5, and the neural network based algorithm, the RX algorithm, in rule extraction process. The results indicated that the two approaches are similar in accuracy. Although the number of rules generated by the neural network approach is less than that generated by the C4.5, the average number of conditions in rules generated by the neural network approach is greater than that of C4.5. The accuracy of the RX rules were nearly the same or slightly greater than those of the network with discrete hidden-unit activation values for the iris dataset. Our future work will modify the RX algorithm in order to improve its performance compared with decision tree based algorithm.

References:

[1] D. Michie, D.J. Spiegelhater, and C.C. Taylor, Machine Learning, Neural and Statistical Classification. Ellis Horwood Series in Artificial Intelligence, 1994.

[2] J.R.Quinlan, C4.5: Programs for Machine learning. Morgan Kaufmann, 1993.

[3] LU,H., SETIONO, R., LIU, H. " Effective Data Mining Using Neural Networks", IEEE Transactions on Knowledge And Data Engineering, Vol. 8, No. 6, pp. 957- 961,1996.

[4] SETIONO, R., "Extracting Rules from Neural Networks by Pruning and Hidden-unit Splitting" Neural Computation, Vol.9, No.1, pp.205-225, 1997.

[5] R.Andrews, J.Diederich, and A.Tickle, "A Survey and critique of techniques for extracting rules from trained artificial neural networks," Knowledge Based Systems, Vol. 8, No. 6, pp. 373-389, 1995.

[6] R. Setiono and H. Liu. "Symbolic representation of neural networks, "IEEE Computer, Vol. 29, No.3, pp. 71-77, 1996.

[7] R. Setiono and H. Liu. "NeuroLinear: From neural networks to oblique decision rules", Neurocomputing, Vol. 17, pp. 1-24, 1997.

[8] C. Merz, and P. Murphy, UCI repository of machine-learning-databases http://www.ics.uci.edu/`mlearn/MLRepository.html, Irvine, 1996.

[9] L. M. Fu, "Rule Generation from Neural Networks," IEEE Transactions on Systems, Man, and Cybernetics, Vol. 24, No. 8, 1994.

[10] R. Setiono, "A Neural Network Construction Algorithm which Maximizes the Likelihood Function," Connection Science, Vol. 7, No. 2, pp. 147-166, 1995.

[11] R. Setiono, "A Penalty Function Approach for Pruning Feed-forward Neural Networks," Neural Computation, Vol. 9, No.1, pp. 301-320, 1997.

[12] Eduardo R.Hurschka and Nelson F.F.Ebecken, " Rule Extraction from Neural Networks: Modified RX Algorithm", IEEE Transactions on Neural Networks, 1999.

[13] R. Setiono and W.K. Leow, "FERNN: An algorithm for fast extraction of rules from neural networks," Applied Intelligence, Vol. 12, No. 1/2, pp. 15-25, 2000.