A Comparative Evaluation Of Symbolic Learning Methods and Neural Learning Methods

Shravya Reddy Konda
Department of Computer Science
University of Maryland, College Park
Email: shravyak@cs.umd.edu

Abstract

In this paper, performance of symbolic learning algorithms and neural learning algorithms on different kinds of datasets has been evaluated. Experimental results on the datasets indicate that in the absence of noise, the performances of symbolic and neural learning methods were comparable in most of the cases. For datasets containing only symbolic attributes, in the presence of noise, the performance of neural learning methods was superior to symbolic learning methods. But for datasets containing mixed attributes (few numeric and few nominal), the recent versions of the symbolic learning algorithms performed better when noise was introduced into the datasets.

1. Introduction

The problem most often addressed by both neural network and symbolic learning systems is the inductive acquisition of concepts from examples [1]. This problem can be briefly defined as follows: given descriptions of a set of examples each labeled as belonging to a particular class, determine a procedure for correctly assigning new examples to these classes. In the neural network literature, this problem is frequently referred to as supervised or associative learning.

For supervised learning, both the symbolic and neural learning methods require the same input data, which is a set of classified examples represented as feature vectors. The performance of both types of learning systems is evaluated by testing how well these systems can accurately classify new examples. Symbolic learning algorithms have been tested on problems ranging from soybean disease diagnosis [2] to classifying chess end games [3]. Neural learning algorithms have been tested on problems ranging from converting text to speech [4] to evaluating moves in backgammon [5].

In this paper, the current problem is to do a comparative evaluation of the performances of the symbolic learning methods which use decision trees such as ID3 [6] and its revised versions like C4.5 [7] against neural learning methods like Multilayer perceptrons [8] which implements a feed-forward neural network with error back propagation.

Since the late 1980s, several studies have been done that compared the performance of symbolic learning approaches to the neural network techniques. Fisher and McKusick [9] compared ID3 and Backpropagation on the basis of both prediction accuracy and the length of training. According to their conclusions, Backpropagation attained a slightly higher accuracy. Mooney et al., [10] found that ID3 was faster than a Backpropagation network, but the Backpropagation network was more adaptive to noisy data sets. Shavlik et al., [1] compared ID3 algorithm with perceptron and backpropagation neural learning algorithms. They found that in all cases, backpropagation took much longer to train but the accuracies varied slightly depending on the type of dataset. Besides accuracy and learning time, this paper investigated three additional aspects of empirical learning, namely, the dependence on the amount of training data, the ability to handle imperfect data of various types and the ability to utilize distributed output encodings.

Depending upon the type of datasets they worked on, some authors claimed that symbolic learning methods were quite superior to neural nets while some others claimed that accuracies predicted by neural nets were far better than symbolic learning methods. The hypothesis being made is that in case of noise free data, ID3 gives faster results whose accuracy will be comparable to that of back propagation techniques. But in case of noisy data, neural networks will perform better than ID3 though the time taken will be more in case of neural networks. Also, in the case of noisy data, performance of C4.5 and neural nets will be comparable since C4.5 too is resistant to noise to an extent due to pruning.

2. Symbolic Learning Methods

In ID3, the system constructs a decision tree from a set of training objects. At each node of the tree the training objects are partitioned by their value along a single attribute. An information theoretic measure is used to select the attribute whose values improve prediction of class membership above the accuracy expected from a random guess. The training set is recursively decomposed in this manner until no remaining attribute improves prediction in a statistically significant manner when the confidence factor is supplied by the user.

So, ID3 method uses Information Gain heuristic which is based on Shannon's entropy to build efficient decision trees. But one disadvantage with ID3 is that it overfits the training data. So, it gives rise to decision trees which are too specific and hence this approach is not noise resistant when tested on novel examples. Another disadvantage is that it cannot deal with missing attributes and requires all attributes to have nominal values.

C4.5 is an improved version of ID3 which prevents over-fitting of training data by pruning the decision tree when required, thus making it more noise resistant.

3. Neural Network Learning Methods

Multilayer perceptron is a layered network comprising of input nodes, hidden nodes and output nodes [11]. The error values are back propagated from the output nodes to the input nodes via the hidden nodes. Considerable time is required to build a neural network but once it is done, classification is quite fast. Neural networks are robust to noisy data as long as too many epochs are not considered since they do not overfit the training data.

4. Evaluation Design

For the evaluation purposes, a free and popular software tool called Weka (Waikato Environment for Knowledge Acquisition) is used. This software has the implementations of several machine learning algorithms made easily accessible to the user with the help of graphical user interfaces.

The training and the test datasets have been taken from the UCI machine learning repository. Two different types of datasets will be used for the evaluation purposes. One type of datasets contain only symbolic attributes (Symbolic Datasets) and the other type contain mixed attributes (Numeric Datasets). Performance of the different learning methods will be evaluated using the original datasets which do not contain any noise and after introducing noise into them. Noise is introduced in the class attributes of the datasets by using the 'AddNoise' filter option in Weka which adds the specified percentage of noise randomly into the datasets.

Symbolic Datasets are those which contain only symbolic attributes. Symbolic learning methods like ID3 and its recent developments can be run only on datasets where all the attributes are nominal. In Weka, these nominal attributes are automatically converted to numeric ones for neural network learning methods. So, preprocessing is not required in this type of datasets.

Numeric Datasets are those which contain few nominal and few numeric attributes. Since symbolic learning methods like ID3 and its recent developments can be run only on datasets where all the attributes are nominal, these datasets first need to be preprocessed. A 'Discretize' filter option available in Weka is used to discretize all the non-symbolic attribute values into individual intervals so that each attribute can now be treated as a symbolic one.

Initially, the entire data being considered is randomized. Two types of evaluation techniques are being used to analyze the data.

(a) Percentage Split: In general, the data will be split up randomly into training data and test data. In the experiments conducted, the data will be split such that training data comprises 66% of the entire data and the rest is used for testing.
(b) K-fold Cross-validation: In general, the data is split into k disjoint subsets and one of it is used as testing data and the rest of them are used as training data. This is continued till every subset has been used once as a testing dataset. In the experiments conducted, 5-fold cross validation was done.

5. Experimental Results

Experiments were conducted on two symbolic datasets and two numeric datasets. The two symbolic datasets are tic-tac-toe and chess. The two numeric datasets are segment and teacher's assistant evaluation (tae).

DataSet 1 : TIC-TAC-TOE

(a) 5-fold cross validation

(i)***Without any noise:***

Classifiers	Time to build	% correct	% incorrect	% not classified
ID3	0.06	86.1169	11.691	2.1921
Multilayer Perceptron	6.35	97.4948	2.5052	0
J48	0.06	85.8038	14.1962	0
C4.5 unpruned	0.01	87.5783	12.4217	0
C4.5 confidence factor = 0.1	0.02	83.1942	16.8058	0

(ii) ***Percentage of noisy data = 10%***

Classifiers	Time to build	% correct	% incorrect	% not classified
ID3	0.03	67.4322	28.0793	4.4885
Multilayer Perceptron	6.16	81.8372	18.1628	0
J48	0.02	75.8873	24.1127	0
C4.5 unpruned	0.06	73.5908	26.4092	0
C4.5 confidence factor = 0.1	0.01	71.2944	28.7056	0

(b) Percentage split with training data being 66% and the rest is testing data

(i)***Without Noise:***

Classifiers	Time to build	% correct	% incorrect	% not classified
ID3	0.05	85.5828	11.0429	3.3742
Multilayer Perceptron	6.5	97.546	2.454	0
J48	0.01	83.1288	16.8712	0
C4.5 unpruned	0.01	88.0368	11.9632	0
C4.5 confidence factor = 0.1	0.02	82.2086	17.7914	0

(ii)***Percentage of Noisy data = 10%***

Classifiers	Time to build	% correct	% incorrect	% not classified
ID3	0.04	68.4049	28.2209	3.3742
Multilayer Perceptron	6.15	80.6748	19.3252	0
J48	0.02	73.9264	26.0736	0
C4.5 unpruned	0.02	72.3926	27.6074	0
C4.5 confidence factor = 0.1	0.01	71.4724	28.5276	0

For the tic-tac-toe dataset, in the presence of noise, neural nets had better prediction accuracies than all the other algorithms as expected. Though C4.5 gives better accuracy than ID3, its accuracy is still lower in comparison to Neural Nets. If the pruning factor (confidence factor was lowered) was increased, the prediction accuracies of C4.5 dropped a little. But in the absence of noise, the performances of ID3 and Multilayer Perceptron should have been comparable. But the performance of Multilayer Perceptron is quite superior to ID3.

DataSet 2 : CHESS

(a) 5-fold cross validation

(i)***Without any noise:***

Classifiers	Time to build	% correct	% incorrect	% not classified
ID3	0.21	99.562	0.438	0
Multilayer Perceptron	47.67	97.4656	2.5344	0
J48	0.15	99.3742	0.6258	0
C4.5 unpruned	0.05	99.3116	0.6884	0
C4.5 confidence factor = 0.1	0.1	99.2178	0.7822	0

(ii) ***Percentage of noisy data = 10%***

Classifiers	Time to build	% correct	% incorrect	% not classified
ID3	0.36	81.1952	18.8048	0
Multilayer Perceptron	47.75	86.796	13.204	0
J48	0.21	89.0488	10.9512	0
C4.5 unpruned	0.18	84.6683	15.3317	0
C4.5 confidence factor = 0.1	0.19	88.4856	11.5144	0

(b) Percentage split with training data being 66% and the rest is testing data

(i)***Without Noise:***

Classifiers	Time to build	% correct	% incorrect	% not classified
ID3	0.13	99.448	0.552	0
Multilayer Perceptron	43.55	97.1481	2.8519	0
J48	0.06	99.08	0.92	0
C4.5 unpruned	0.06	98.988	1.012	0
C4.5 confidence factor = 0.1	0.08	99.08	0.92	0

(ii)***Percentage of Noisy data = 10%***

Classifiers	Time to build	% correct	% incorrect	% not classified
ID3	0.33	80.1288	19.8712	0
Multilayer Perceptron	41.73	85.7406	14.2594	0
J48	0.24	87.5805	12.4195	0
C4.5 unpruned	0.19	82.6127	17.3873	0
C4.5 confidence factor = 0.1	0.19	87.6725	12.3275	0

For the chess dataset, in the absence of noise, the performance of ID3 is better than that of Multilayer perceptron and takes lesser time. For the noisy data, back propagation predicts better accuracies than that of ID3 as expected, but the performance of C4.5 is slightly higher than back propagation. The reason for this could be that the feature space in this dataset is more relevant. So, C4.5 builds a tree and prunes it to get a more efficient tree.

DataSet 3 : SEGMENT

(a) 5-fold cross validation

(i) ***Without any noise:***

Classifiers	Time to build	% correct	% incorrect	% not classified
ID3	0.05	88.0667	5.2	6.7333
Multilayer Perceptron	10.3	90.6	9.4	0
J48	0.02	91.6	8.4	0
C4.5 unpruned	0.23	94	6	0
C4.5 confidence factor = 0.1	0.12	94.3333	5.6667	0

(ii) ***Percentage of noisy data = 10%***

Classifiers	Time to build	% correct	% incorrect	% not classified
ID3	0.07	68.9333	21.3333	9.7333
Multilayer Perceptron	9.64	80.8667	19.1333	0
J48	0.04	81.2667	18.7333	0
C4.5 unpruned	0.04	79.6	20.4	0
C4.5 confidence factor = 0.1	0.03	80.5333	19.4667	0

(b) Percentage split with training data being 66% and the rest is testing data

(i) ***Without Noise:***

Classifiers	Time to build	% correct	% incorrect	% not classified
ID3	0.06	89.8039	4.1176	6.0784
Multilayer Perceptron	9.87	87.6471	12.3529	0
J48	0.03	92.1569	7.8431	0
C4.5 unpruned	0.02	93.7255	6.2745	0
C4.5 confidence factor = 0.1	0.03	90.1961	9.8039	0

(ii) ***Percentage of Noisy data = 10%***

Classifiers	Time to build	% correct	% incorrect	% not classified
ID3	0.07	72.9412	19.6078	7.451
Multilayer Perceptron	11.73	82.549	17.451	0
J48	0.03	82.1569	17.8431	0
C4.5 unpruned	0.04	82.549	17.451	0
C4.5 confidence factor = 0.1	0.03	81.3725	18.6275	0

Segment, being a numeric dataset, all the attribute values had to be discretized before running the algorithms. In the absence of noise, ID3 performs slightly better than back propagation and the performance of J48 (implementation of C4.5 in Weka) is much better than ID3 and backpropagation. But a very interesting observation was found. In the absence of noise, the performance of an unpruned tree generated by C4.5 was quite superior to the rest. In the presence of noise, the performances of back propagation and C4.5 were comparable.

DataSet 4 : TAE

(a) <u>5-fold cross validation</u>

(i) ***Without any noise:***

Classifiers	Time to build	% correct	% incorrect	% not classified
ID3	0.02	54.3046	35.0993	10.596
Multilayer Perceptron	0.18	54.9669	45.0331	0
J48	0.02	48.3444	51.6556	0
C4.5 unpruned	0.01	50.9934	49.0066	0
C4.5 confidence factor = 0.1	0.01	47.0199	52.9801	0

(ii) ***Percentage of noisy data = 10%***

Classifiers	Time to build	% correct	% incorrect	% not classified
ID3	0.02	53.6424	37.0861	0
Multilayer Perceptron	0.16	38.4106	61.5894	0
J48	0.02	52.9801	47.0199	0
C4.5 unpruned	0.01	56.2914	43.7086	0
C4.5 confidence factor = 0.1	0.01	54.3046	45.6954	0

(b) <u>Percentage split with training data being 66% and the rest is testing data</u>

(i) ***Without Noise:***

Classifiers	Time to build	% correct	% incorrect	% not classified
ID3	0.02	44.2308	34.6154	21.1538
Multilayer Perceptron	2.23	57.6923	42.3077	0
J48	0.03	51.9231	48.0769	0
C4.5 unpruned	0.02	55.7692	44.2308	0
C4.5 confidence factor = 0.1	0.01	42.3077	57.6923	0

(ii) ***Percentage of Noisy data = 10%***

Classifiers	Time to build	% correct	% incorrect	% not classified
ID3	0.01	38.4615	40.3846	21.1538
Multilayer Perceptron	0.17	44.2308	55.7692	0
J48	0.01	44.2308	55.7692	0
C4.5 unpruned	0.01	50	50	0
C4.5 confidence factor = 0.1	0.01	44.2308	55.7692	0

TAE, being a numeric dataset, its attribute values had to be discretized too before running the algorithms. But after observing the results, it is very clear that the random discretization provided by Weka did not generate good intervals due to which the overall accuracy predicted by all the methods is quite poor. Again, interestingly an unpruned tree built by C4.5 seems to give high prediction accuracies relative to the rest in most of the cases. In this case, for cross-validation approach and noisy data, surprisingly the performance of back-propagation was very poor. One reason for this could be that only few epochs of the training data were run to build the neural network. In the absence of noise, accuracy prediction of Multilayer perceptron was either comparable or greater than that of ID3.

6. Conclusion

No single machine learning algorithm can be considered superior to the rest. The performance of each algorithm depends on what type of dataset is being considered, whether the feature space is relevant and whether the data contains noise. In the absence of noise, in some cases, the performance of ID3 was comparable or sometimes better than back-propagation and was faster but in some cases Multilayer perceptron performed better. When noisy datasets were considered, back propagation definitely did better than ID3 though it took more time to build the neural network. But in the presence of noise, in some cases, C4.5 gave faster and better results when the attributes being considered were relevant. But some surprising observations were made when the attribute values of the numeric datasets were discretized, the prediction accuracy of an unpruned tree generated by C4.5 algorithm was much higher than the rest. This shows that the unpruned tree generated by C4.5 is not the same as that generated by ID3.

References:

1.Mooney, R., Shalvik, J., and Towell, G. (1991): Symbolic and Neural Learning Algorithms - An experimental comparison, in *Machine Learning 6*, pp. 111-143.

2. Michalski, R.S., & Chilausky, R.L. (1980): Learning by being told and learning from examples - An experimental comparison of two methods of knowledge acquisition in the context of developing an expert system for soybean disease diagnosis, in *Policy Analysis and Information Systems, 4,* pp. 125-160.

3. Quinlan, J.R. (1983): Learning efficient classification procedures and their application to chess end games in R.S. Michalski, J.G. Carbonell, & T.M. Mitchell (Eds.), in *Machine learning: An artificial intelligence approach* (Vol. 1). Palo Alto, CA: Tioga.

4. Sejnowski, T.J., & Rosenberg, C. (1987): Parallel networks that learn to pronounce English text, in *Complex Systems,* 1, pp. 145-168.

5. Tesauro, G., & Sejnowski, T.J. (1989): A parallel network that learns to play backgammon, in *Artificial Intelligence, 39*, pp. 357-390.

6. Quinlan, J.R. (1986): Induction on Decision Trees, in *Machine Learning 1*, 1

7. Quinlan, J.R. (1993): C4.5 – Programs for Machine Learning. San Mateo: Morgan Kaufmann.

8. Rumelhart, D., Hinton, G., & Williams, J. (1986): Learning Internal Representations by Error Propagation, in *Parallel Distributed Processing, Vol. 1* (D. Rumelhart *k* J. McClelland, eds.). MIT Press.

9. Fisher, D.H. and McKusick, K.B. (1989): An empirical comparison of ID3 and backpropagation, in *Proc. of the Eleventh International Joint Conference on Artificia1 Intelligence (IJCAI-89)*, Detroit, MI, August 20-25, pp. 788-793.

10. Mooney, R., Shavlik, J., Towell, G., and Gove, A.(1989): An experimental comparison of symbolic and connectionist learning algorithms, in *Proc. of the Eleventh International Joint Conference on Artificial Intelligence (IJCAI-89),* Detroit, MI, August 20-25, pp. 775-780**.**

11. McClelland, J. k Rumelhart, D. (1988). Explorations in Parallel Distributed Processing, MIT Press, Cambridge, MA.

.