Available online at www.sciencedirect.com

ScienceDirect

ELSEVIER

Procedia Computer Science

Procedia Computer Science 20 (2013) 433 – 438

Complex Adaptive Systems, Publication 3
Cihan H. Dagli, Editor in Chief
Conference Organized by Missouri University of Science and Technology
2013- Baltimore, MD

Knowledge Extraction from Survey Data Using Neural Networks

Imran Khan, Arun Kulkarni

The University of Texas at Tyler, Tyler, TX 75799, USA

Abstract

Surveys are an important tool for researchers. It is increasingly important to develop powerful means for analyzing such data and to extract knowledge that could help in decision-making. Survey attributes are typically discrete data measured on a Likert scale. The process of classification becomes complex if the number of survey attributes is large. Another major issue in Likert-Scale data is the uniqueness of tuples. A large number of unique tuples may result in a large number of patterns. The main focus of this paper is to propose an efficient knowledge extraction method that can extract knowledge in terms of rules. The proposed method consists of two phases. In the first phase, the network is trained and pruned. In the second phase, the decision tree is applied to extract rules from the trained network. Extracted rules are optimized to obtain a comprehensive and concise set of rules. In order to verify the effectiveness of the proposed method, it is applied to two sets of Likert scale survey data, and results show that the proposed method produces rule sets that are comparable with other knowledge extraction techniques in terms of the number of rules and accuracy.

Keywords: Neural networks; Decision trees; Rule extraction; Knowledge discovery; Classification

1. Introduction

A survey is conducted to collect data from individuals to find out their behaviors, needs and opinions towards a specific area of interest. Survey responses are then transformed into usable information in order to improve or enhance that area. Survey data attributes can come in the forms of binary-valued (or binary-encoded), continuous data or discrete data measured on a Likert scale. All three forms of data attributes are used according to the survey requirements. Discrete data can be used as a measure on a Likert scale to provide some distinct advantages over the other two types of data attributes. It helps respondents choose an answer. For instance, some respondents may be too impatient to make fine judgments and to give their responses on a continuous scale. The options provided in a typical five-level Likert item are Strongly Disagree, Disagree, neither Agree nor Disagree, Agree and Strongly Agree. The collected data might be contaminated if the difficult or time consuming judgmental task is beyond the respondent's ability or tolerance. The use of a Likert scale has been proposed to alleviate these difficulties.

Classification and knowledge extraction from survey data is a very important step in the decision-making process. Based on this knowledge, decisions are taken to improve the area for which the survey was conducted. Classification of Likert-scale survey data depends on the number of attributes. Classification process may become more complex when the number of Likert scale options and attributes in the survey is large. In the case of a survey, these attributes or features are the questions. Another major issue in Likert-Scale data is the uniqueness of the tuples. Classification algorithms group data based on the patterns of the attributes. A large number of unique tuples may result in a large number of patterns. Due to a large number of patterns, the knowledge extraction process from these classifiers becomes complex, and often the outcome of knowledge extraction process may not be satisfactory. The main focus of this research is to classify Likert-scale survey data using a multi-layered feed forward (MLF) [1,2,3,4] neural network and to apply Artificial Neural Network Tree (ANNT) algorithm [5,6] to extract knowledge from trained neural network.

The method proposed in this research consists of two steps. The first step is to train and prune the MLP neural network using back propagation algorithm. The second step is to apply an ANNT algorithm to extract knowledge from the neural network in the form of rules and optimize them to obtain a comprehensive and concise set of rules.

doi:10.1016/j.procs.2013.09.299

The proposed method was applied to two Likert scale surveys. The first survey was about the reading strategies of students. The name of the survey was "Metacognitive Awareness of Reading Strategies Inventory (MARSI)" [7]. The second data set is a teacher evaluation survey. The teacher evaluation survey is used to evaluate a teacher's performance and helped in decision making.

2. Method

Method to extract the knowledge from Likert scale survey data consists of two steps. The first step is to train and prune the neural network using a multi-layered back propagation algorithm. The second step is to apply an ANNT algorithm to extract rules from trained network. Responses of a Likert-scale survey are usually in a non-numeric form. For neural network training, responses were converted to the range of 1 to -1. The mapping shown in Table 1 was used.

Table 1. Normalization of Responses

Option	Option Value	Normalized Value
Option 1	1	-0.9
Option 2	2	-0.4
Option 3	3	-0.1
Option 4	4	0.4
Option 5	5	0.9

2.1 Neural network training and pruning

A MLF neural network consists of three layers (Figure 1). The first layer has *k* input neurons which send data via connection links to the second layer of *M* hidden neurons, and then via more connection links to the third layer of output neurons. The number of neurons in the input layer is usually based on the number of features in a data set. The second layer is also called the hidden layer. More complex systems will have multiple hidden layers of neurons. Given an input pattern x_i, $i \in \{1,2,\ldots k\}$, where k is the number of attributes in the data set, the activation value of each neuron *o* can be described by the following equation:

$$o_i = f\left(\sum_j (w_{ij} \,.\, x_j)\right) \tag{1}$$

where *f*(.) is the activation function. In this research, sigmoid function is used.

$$f(net) = \frac{1}{1 + exp^{-net}} \tag{2}$$

In order to calculate the change of weights, output vector **o** is compared with the target vector **d**, and the error between the two vectors is then propagated backward to obtain the change in weights Δq_{ij} that is used to update the weights. Δq_{ij} for weights between layers L_2L_3 is given by:

$$\Delta q_{ij} = \alpha \delta_i o_j \tag{3}$$

where α is a training rate coefficient (typically 0.01 to 1.0). o_j is the output of neuron *j* in layer L_3, and δ_i is given by

$$\delta_i = (d_i - o_i)o_i(1 - o_i) \tag{4}$$

where o_i represents the actual output, where as d_i represents the target output.

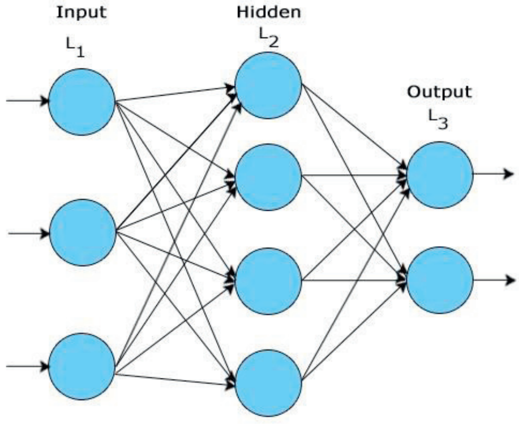

Figure 1. Three Layer Artificial Neural Network.

The back-propagation algorithm trains the hidden layers by propagating the output error back through layer by layer, adjusting weights at each layer. The change in weights Δp_{ij} between layers L1L2 can be calculated as

$$\Delta p_{ij} = -\beta o_j \delta_{Hi} \tag{5}$$

where β is a training rate coefficient (typically 0.01 to 1.0). o_j is the output of neuron j in layer L2, and δ_{Hi} is given by

$$\delta_{Hi} = o_i(1 - o_i) \sum_{k=1}^{m} \delta_k q_{ik} \tag{6}$$

where o_i is the output of neuron i in layer L2, and the summation term represents the weighted sum of all values corresponding to neurons in layer L3 that are obtained by using Equation (3) .

Pruning techniques help in reducing the size of the network that results in a reduction of processing time and complexity. In order to prune the network, the first step is to determine the optimal number of neurons in the hidden layer. To determine the number of neurons in the hidden layer, we started with a large number of hidden units and trained the network to find the classification accuracy. The original size of the hidden layer is then reduced by removing one node at a time and retraining the network to find the accuracy. If the accuracy is dropped below the minimum acceptable accuracy then the earlier network configuration is restored otherwise the unit will be considered as redundant. This process will be repeated for each node in the hidden layer until the optimal number of neurons is determined [8]. We have applied the same approach to remove redundant and irrelevant input units.

2.2 Rules Extraction

The trained knowledge-based network is used for rule generation in if-then form in order to justify any decision reached. These rules describe the extent to which a test pattern belongs or does not belong to one of the classes in terms of antecedent and consequent clauses. We have applied an ANNT algorithm in order to extract rules from trained neural network. For ANNT illustration, Teacher Evaluation survey data have been used. A network with five features and two hidden units is trained to classify data into two classes, i.e. satisfied students and dissatisfied students. The ANNT algorithm is described below.

Step 1: From the trained and pruned network, build a decision tree using the weights and activation patterns of hidden and output layer. Extract the intermediate rules from the hidden-output tree. C4.5 algorithm [9] has been used in this research to build the decision tree. Figure 2 illustrates this step.

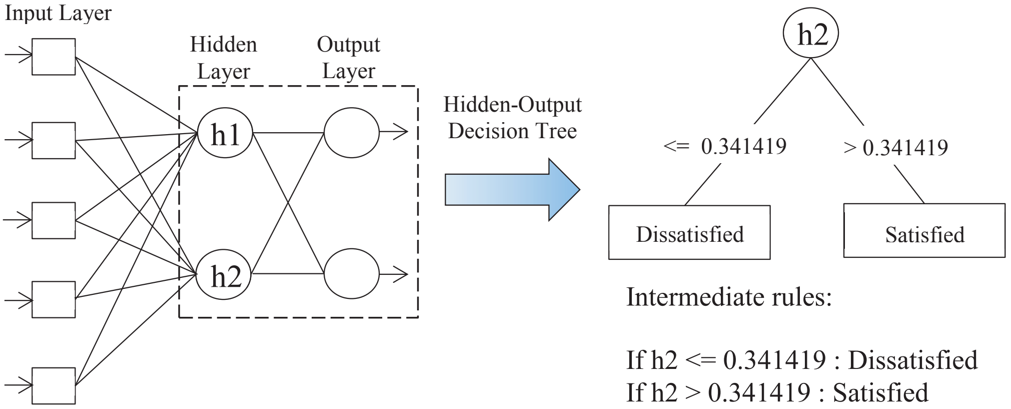

Figure 2. Illustration of Step 1 of ANNT algorithm.

Step 2: Build input-hidden decision trees for each hidden unit that is part of the intermediate rule generated in Step 1. In this case input-hidden decision tree will be generated for only h2. Extract the input rules from input-hidden tree. Figure 3 illustrates this step.

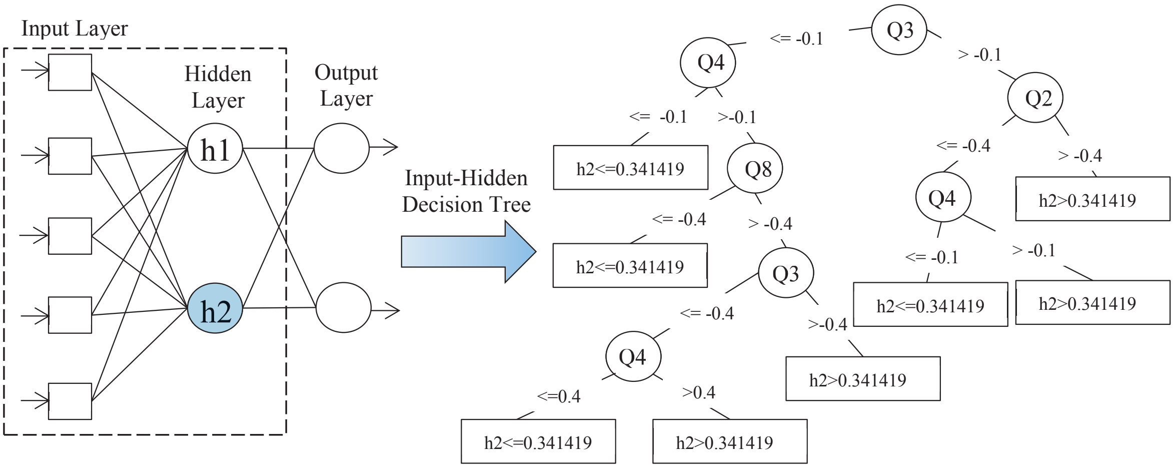

Figure 3. Illustration of Step 2 of ANNT algorithm.

Step 3: Obtain final rules by substituting the input rules in the intermediate rules. For illustration, one rule is shown in Figure 4.

If Q3 in (OPT4, OPT5) and Q2 in (OPT3, OPT4, OPT5) Then Satisfied

Figure 4. Final rule

Step 4: Rule pruning includes removing redundant rules, replacing specific rules with more general rules, and merging of rules.

3. Experiments

As an illustration, this research has applied the method to two different survey data sets. The first survey is about reading strategies for students, and the second survey is regarding teacher evaluation. To compare the efficiency of this proposed method, C4.5 has been applied to the same data sets using two different methods: K-fold cross validation and split-sample. The outcome of C4.5 is then compared to the results of the method applied in this research.

3.1 MARSI Survey Data

MARSI was developed to assess a student's reading awareness. It has 30 questions, and each question has five-level Likert options. These 30 questions described 30 strategies or actions readers use when reading book chapters, articles etc. Data is grouped into three classes: "High Level of Awareness", "Medium Level of Awareness" and "Low Level of Awareness". A total of 877 students participated in this survey but after data cleaning, 860 records were selected for analysis. C4.5 algorithm has been applied using two methods: K-fold cross validation and split-sample. For split-sample method, 50% of the data samples have been used for training and the remaining 50% has been used for testing. For K-fold cross validation method, data was divided into 10 subsets of approximately equal size. A comparison of ANNT and C4.5 rule extraction algorithms is shown in Table 2. Following are the top three rules extracted from MARSI survey using ANNT algorithm. The numbers in parentheses represent the number of samples classified by that rule.

Rule 1: IF Q20 in (OPT3,OPT4,OPT5) And Q11 in (OPT4,OPT5) And Q15 in (OPT3,OPT4,OPT5)
And Q17 in (OPT3,OPT4,OPT5) And Q13 in (OPT4,OPT5) And Q30 in (OPT4,OPT5)
THEN "High Level of Awareness" (326)
Rule 2: IF Q20 in (OPT2,OPT1) And Q29 in (OPT3,OPT4,OPT5) And Q12 in (OPT3,OPT4,OPT5) And Q15 in (OPT1,OPT2)
THEN "Medium Level of Awareness" (52)
Rule 3: IF Q20 in (OPT3,OPT4,OPT5) And Q11 in (OPT4,OPT5) And Q15 in (OPT3,OPT4,OPT5)
And Q17 in (OPT3,OPT4,OPT5) And Q13 in (OPT1,OPT2,OPT3) And Q18 in (OPT4,OPT5) And Q6 in (OPT4,OPT5)
And Q12 in (OPT2,OPT3,OPT4,OPT5)
THEN "High Level of Awareness" (31)

Table 2. Comparison of Different Rule Extraction Techniques for MARSI Survey Data

Rule Extraction Technique	Number Of Rules	Performance Accuracy
ANNT	55	88.13%
C4.5 (K-fold cross validation)	56	77.21%
C4.5 (split-samples)	55	76.75%

3.2 Teacher Evaluation Survey Data

The teacher evaluation survey contained 8 questions; each question has five-level Likert options. Data is grouped into two classes: "Satisfied Students" and "Dissatisfied Students". A total of 265 students participated in this survey. For split-sample method, 50% of the data samples have been used for training and the remaining 50 % has been used for testing. For K-fold cross validation method, data was divided into 10 subsets of approximately equal size. A comparison of ANNT and C4.5 rule extraction algorithms is shown in Table 3. Following are the top three rules extracted from the teacher evaluation survey using ANNT algorithm.

Rule 1: IF Q3 in (OPT4, OPT5) And Q2 in (OPT3, OPT4, OPT5)
THEN "Satisfied Students" (155)
Rule 2: IF Q3 in (OPT3, OPT2, OPT1) And Q4 in (OPT3, OPT2, OPT1)
THEN "Dissatisfied Students" (60)
Rule 3: IF Q3 in (OPT3, OPT2, OPT1) And Q4 in (OPT4, OPT5) And Q8 in (OPT2, OPT1)
THEN "Dissatisfied Students" (16)

Table 3. Comparison of Different Rule Extraction Techniques for Teacher Evaluation Survey Data

Rule Extraction Technique	Number Of Rules	Performance Accuracy
ANNT	8	95.84%
C4.5 (K-fold cross validation)	11	87.92%
C4.5 (split-samples)	11	87.92%

4. Conclusion

The effectiveness of the proposed method was tested by applying it to two different surveys having discrete data measured on a Likert scale. The first survey had a large number of attributes with a large number of unique patterns. The second survey had fewer attributes with fewer unique patterns. The quality of rules extracted using ANNT and C4.5 algorithm can be measured by accuracy, comprehensibility and fidelity. From the experimental results shown in Table 2 and Table 3, it can be stated that the rules extracted for these two data sets using ANNT algorithm has a higher accuracy as compared to C4.5 algorithm. Comprehensibility of ANNT method is high or comparable because the number of extracted rules is low as compare to C4.5

While the proposed method can be expected to perform well in general, it suffers from some limitations as well. This method assumes that all the attributes in the given data sets are discrete data measured on a Likert scale. The proposed method may require preprocessing of the data with non-discrete attributes. The viability of this method should be tested on a wide variety of Likert-scale data and also compared the results with C5.0 algorithm.

References

1. G. E. Hinton, "How neural networks learn from experience," in *Mind and brain: Readings from Scientific American magazine*, ed New York, NY US: W H Freeman/Times Books/ Henry Holt & Co, 1992, pp. 113-124.
2. J. Hopfield and D. Tank, "'Neural' computation of decisions in optimization problems," *Biological Cybernetics,* vol. 52, p. 141, 1985.
3. R. P. Lippman, "An introduction to computing with neural nets". *IEEE ASSP Magazine,* vol. 3 No. 4, pp. 4-22, 1987.
4. S. S. Haykin, *Neural networks : a comprehensive foundation / Simon Haykin*: Upper Saddle River, N.J. : Prentice Hall, c1999.
5. S. K. Anbananthen, G. Sainarayanan, A. Chekima, and J. Teo, "Data Mining using Pruned Artificial Neural Network Tree (ANNT)," *2nd International Conference on Information& Communication Technologies,* p. 1350, Jan. 2006.
6. S. Kalaiarasi, S. Sayeed, and J. Hossen, "Comparison Of Network Pruning And Tree Pruning On Artificial Neural Network Tree," *Australian Journal of Basic & Applied Sciences,* vol. 5, pp. 1093-1098, 2011.
7. Mokhtari, K. and Reichard, C. Assessing students' metacognitive awareness of reading strategies. *Journal of Educational Psychology,* vol 94 No. 2, pp. 249-259, 2002.
8. Hagiwara M (1994). A simple and effective method for removal of hidden units and weights. Neurocomputing 6: 207-218.
9. Quinlan J. *C4.5: Programs for Machine Learning*. Morgan Kaufmann. 1993.