

Discovering Numerical-Type Dependencies for Improving the Accuracy of Decision Trees

*Yue-Shi Lee, **Show-Jane Yen and *Chen-Wei Fang

*Department of Information Management, Ming Chuan University
5 The-Ming Rd., Gwei Shan District, Taoyuan County 333, Taiwan, R.O.C.
Phone: +886-2-27312296, Fax: +886-3-3294449, E-mail. leeys@mcu.edu.tw

** Dept. of Computer Science and Information Management, Fu Jen Catholic University
510 Chung Cheng Rd., Hsinchuan, Taipei 242, Taiwan, R.O.C.
Phone: +886-3-3173312, Fax: +886-2-29023550, E-mail. sjyen@csie.fju.edu.tw

Abstract

As we know, the decision tree learning algorithms, e.g., C5, are good at dataset classification. But those algorithms usually work with only one attribute at a time. The dependencies among attributes are not considered in those algorithms. Unfortunately, in the real world, most databases contain attributes, which are dependent. Thus, it is very important to construct a model to discover the dependencies among attributes, and to improve the accuracy and effectiveness of the decision tree learning algorithms. Neural network model is a good choice for us to concern with the problems of attribute dependencies. Generally, these dependencies are classified into two types: categorical-type dependency and numerical-type dependency. This paper focuses on the numerical-type dependency and proposes a Neural Decision Tree (NDT) model, to deal with such kind of dependencies. The NDT model combines the neural network technologies and the traditional decision-tree learning capabilities, to handle the complicated and real cases. According to the experiments on five datasets from the UCI database repository, the NDT model can significantly improve the accuracy and effectiveness of C5.

Keywords: Attribute Dependency, Data Mining, Decision Tree, Neural Network

Contact Author: Yue-Shi Lee

***This paper is submitted to the Workshop on Artificial Intelligence.

Discovering Numerical-Type Dependencies for Improving the Accuracy of Decision Trees

*Yue-Shi Lee, **Show-Jane Yen and *Chen-Wei Fang

*Department of Information Management, Ming Chuan University, Taoyuan, Taiwan, R.O.C.

**Dept. of Computer Science and Information Engineering, Fu Jen Catholic University, Taipei, Taiwan, R.O.C.

E-mail. leeys@mcu.edu.tw, sjyen@csie.fju.edu.tw

Abstract

As we know, the decision tree learning algorithms, e.g., C5, are good at dataset classification. But those algorithms usually work with only one attribute at a time. The dependencies among attributes are not considered in those algorithms. Unfortunately, in the real world, most databases contain attributes, which are dependent. Thus, it is very important to construct a model to discovery the dependencies among attributes, and to improve the accuracy and effectiveness of the decision tree learning algorithms. Neural network model is a good choice for us to concern with the problems of attribute dependencies. Generally, these dependencies are classified into two types: categorical-type dependency and numerical-type dependency. This paper focuses on the numerical-type dependency and proposes a Neural Decision Tree (NDT) model, to deal with such kind of dependencies. The NDT model combines the neural network technologies and the traditional decision-tree learning capabilities, to handle the complicated and real cases. According to the experiments on five datasets from the UCI database repository, the NDT model can significantly improve the accuracy and effectiveness of C5.

Keywords: Attribute Dependency, Data Mining, Decision Tree, Neural Network

1. Introduction

One important application of data mining is the ability to perform classification in a huge amount of data [1]. The decision-tree learning algorithm is one of the most important results [2]. The decision-tree learning algorithms derive decision trees and rules based on the training dataset. These trees and rules can be used to predict the classes of new instances. Basically, they build decision trees by recursively partitioning the dataset according to the selected attributes. In this framework, they usually deal with only one attribute at a time. The dependencies among attributes are not considered in those algorithms. Unfortunately, in the real world, most datasets contain attributes, which are dependent. Thus, it is very important to construct a model to discover the dependencies among attributes, and to improve the accuracy of the decision tree algorithms.

Neural network model is a good choice for us to concern with the problems of attribute dependencies. Generally, these dependencies are classified into two types: categorical-type dependency and numerical-type dependency. This paper focuses on the numerical-type dependency and proposes a Neural Decision Tree (NDT) model, to deal with such kind of dependencies. The NDT model combines the neural network technologies and the traditional decision-tree learning capabilities to handle the complicated and real cases.

This paper is organized as follows. Section 2 introduces numerical-type dependency. Section 3 then describes our NDT model. Before concluding, the experimental results are demonstrated in Section 4.

2. Numerical-Type Dependency

To describe what is the numerical-type dependency, Table 1 shows a numerical-type dataset, which is a stoke price dataset with 18 instances.

Table 1 Stock Price Dataset with 18 Instances

Price on Day T	Forecast Price on Day T+1	Target: Decision
17.60	17.72	Buy
17.70	17.60	Sell
17.70	17.71	Buy
17.71	17.94	Buy
17.72	17.70	Sell
17.75	17.84	Buy
17.84	17.97	Buy
17.90	17.75	Sell
17.92	18.09	Buy
17.94	18.08	Buy
17.97	18.08	Buy
18.02	17.92	Sell
18.08	17.90	Sell
18.08	18.10	Buy
18.08	18.16	Buy
18.09	18.08	Sell
18.10	18.11	Buy
18.16	18.02	Sell

It is generated by the following two simple rules [3].

IF the stock price on date T is greater than the forecast price on date T+1, sell it.

IF the stock price on date T is less than the forecast price on date T+1, buy it.

Based on the C5 and Table 1, no rules can be generated. After we remove the 17th instance, C5 obtains four rules, which are listed below. The relationships between dataset and rules are shown in Figure 1.

If Forecast Price on Day T+1 > 18.02, Buy the Stock.

If Forecast Price on Day T+1 <= 18.02 and Price on Day T > 17.84, Sell the Stock.

If Forecast Price on Day T+1 \leq 18.02, Price on Day T \leq 17.84 and
Forecast Price on Day T+1 $>$ 17.7, Buy the Stock.

If Forecast Price on Day T+1 \leq 18.02, Price on Day T \leq 17.84 and
Forecast Price on Day T+1 \leq 17.7, Sell the Stock.

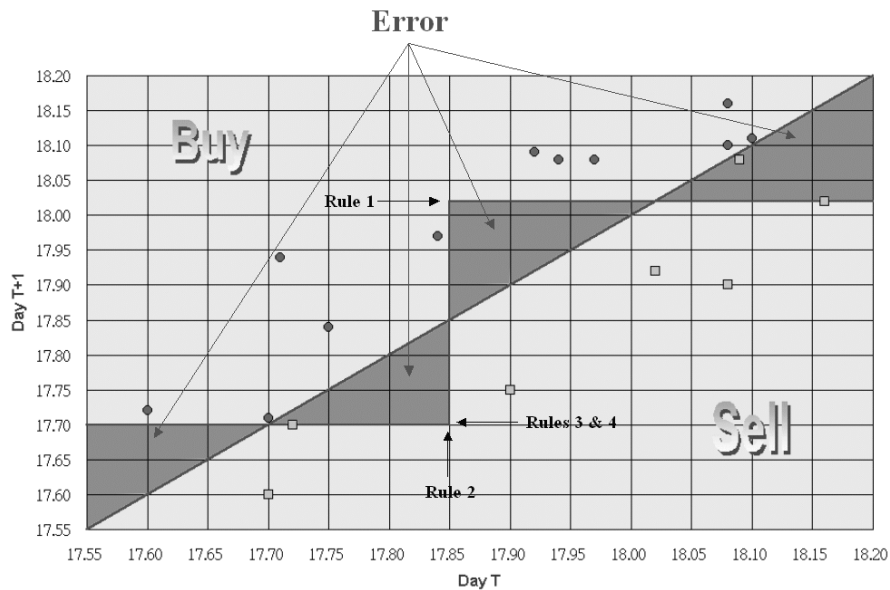


Figure 1 The Relationship between Dataset and Rules

According to the four induction rules listed above, we will make wrong prediction if the data fell into the four darker areas. The above example shows that even a simple rule “If Forecast Price on Day T+1 is greater than Price on Day T, Buy the Stock.” cannot be correctly generated by C5. This is because the numerical-type dependencies are not considered in the traditional decision-tree learning algorithms, e.g., C5. In next section, we will describe our NDT model for discovering such kind of dependencies.

3 Neural Decision Tree (NDT) Model

The architecture of our Neural Decision Tree (NDT) model is depicted in Figure 2.

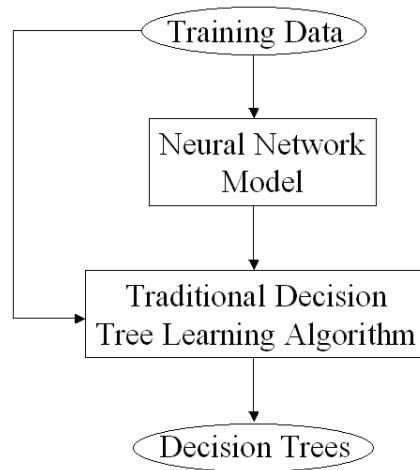


Figure 2 The architecture of the NDT model

In this architecture, the training data is firstly sent to the neural network model. Artificial neural networks were inspired from biology [4]. It has been applied to many applications for different purposes [3, 4, 5]. In this paper, the neural network is used to find the dependencies among attributes. The model used in this paper is the Back-Propagation (BP) model. The training data and the results obtained by the neural network model are then sent to the traditional decision-tree learning algorithm, i.e., C5, to improve the accuracy of C5 and generate a more compact decision tree.

The followings illustrate the rule extraction steps in NDT model over the numerical-categorical-mixed dataset.

1. We separate the numerical-categorical-mixed dataset into two parts, e.g., numerical subset and nominal subset.
2. For categorical subset, we do nothing for it, currently.
3. For numerical subset, we firstly normalize each attribute of the dataset for training a neural network. This is because the neural network model only accepts numerical data as input. We adopt min-max normalization method [6] for each input attribute. Because the squashing sigmoid function, which we adopt in the neural network model, cannot exactly reach the target value 0 or 1, we adopt

min-max normalization method with target value 0.2 and 0.8, instead of 0 and 1, for each output attribute [7]. Based on the normalized dataset, we then train a back-propagation neural network, collect weights between the input layer and the first hidden layer, and change each attribute value according to these weights.

4. We combine the categorical subset and the new numerical subset into a new numerical-categorical-mixed dataset.
5. The new dataset is sent to the C5 to generate the decision tree and rules.

To clearly describe these steps, the followings will describe the details about how to infer the decision tree and rules for Table 1.

In Table 1, there are two numerical-type input attributes and one categorical-type output attribute with two targets (Buy or Sell). First of all, we prepare this dataset by normalizing each input attribute using min-max normalization method. The equation is listed below.

$$V_{\text{new}} = \frac{V_{\text{old}} - 17.6}{0.56}$$

For categorical-type output attribute, we also adopt min-max normalization method with target value 0.2 and 0.8 instead of 0 and 1. The normalized dataset is shown in Table 2.

According to this normalized dataset, we then train a neural network model with the following parameters: Hidden Layer: 1, Input Nodes: 2, Hidden Nodes: 2, Output Nodes: 2, Number of Instances: 18, Training Cycles: 30,000, Initial Weight Bound: 0.3, Learning Rate: 1.0, Decreased by: 0.95, Lower Bound: 0.1, and Momentum: 0.5. Based on these parameters, the training results are shown in Figure 3.

Table 2 Normalized Stock Price Dataset

Price on Day T	Forecast Price on Day T+1	Target: Decision	
		Buy	Sell
0	0.214	0.8	0.2
0.179	0	0.2	0.8
0.179	0.196	0.8	0.2
0.196	0.607	0.8	0.2
0.214	0.179	0.2	0.8
0.268	0.429	0.8	0.2
0.429	0.661	0.8	0.2
0.536	0.268	0.2	0.8
0.571	0.875	0.8	0.2
0.607	0.857	0.8	0.2
0.661	0.857	0.8	0.2
0.750	0.571	0.2	0.8
0.857	0.536	0.2	0.8
0.857	0.893	0.8	0.2
0.857	1	0.8	0.2
0.875	0.857	0.2	0.8
0.893	0.911	0.8	0.2
1	0.750	0.2	0.8

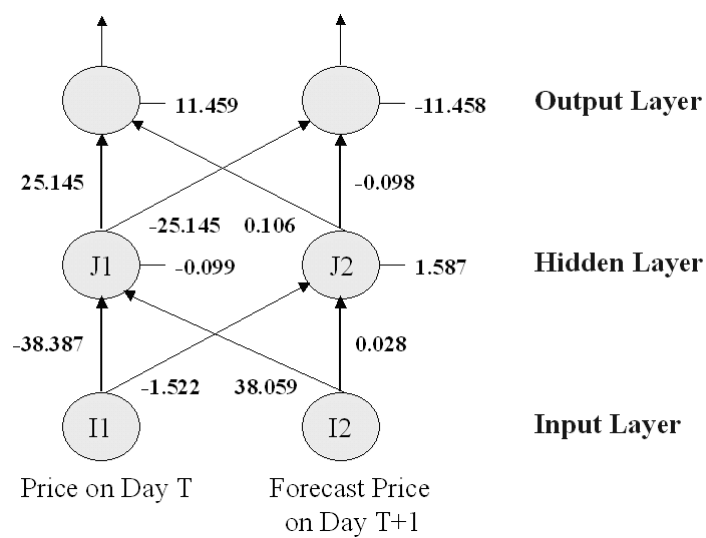


Figure 3 Training Results for Normalized Stock Price Dataset

We examine the link weights between the input layer and the first hidden layer. Then, we change the attribute values according to these weights. That is, we transform the normalized stock price dataset into a new one by the following formula.

$$J1 = (-38.387) * I1 + (38.059) * I2 - (-0.099)$$

$$J2 = (-1.522) * I1 + (0.028) * I2 - (-1.587)$$

I1 means “Normalized Price on Day T”, I2 means “Normalized Forecast Price on Day T+1”, J1 means “New Price on Day T” and J2 means “New Forecast Price on Day T+1”. The transformed results generated by this way are shown in Table 3.

Table 3 Transformed Stock Price Dataset

Price on Day T	Forecast Price on Day T+1	Target: Decision
8.057	1.593	Buy
-6.954	1.315	Sell
0.522	1.321	Buy
15.468	1.305	Buy
-1.529	1.266	Sell
5.930	1.191	Buy
8.596	0.953	Buy
-10.469	0.779	Sell
11.267	0.742	Buy
9.217	0.687	Buy
7.160	0.605	Buy
-7.141	0.462	Sell
-12.613	0.297	Sell
0.979	0.307	Buy
5.057	0.310	Buy
-1.066	0.279	Sell
0.288	0.254	Buy
-9.942	0.086	Sell

Then, the transformed dataset is sent to the C5 to generate the decision tree and rules. Using the original and the transformed datasets, the results generated by the C5 are shown in Figures 4 and 5, respectively.

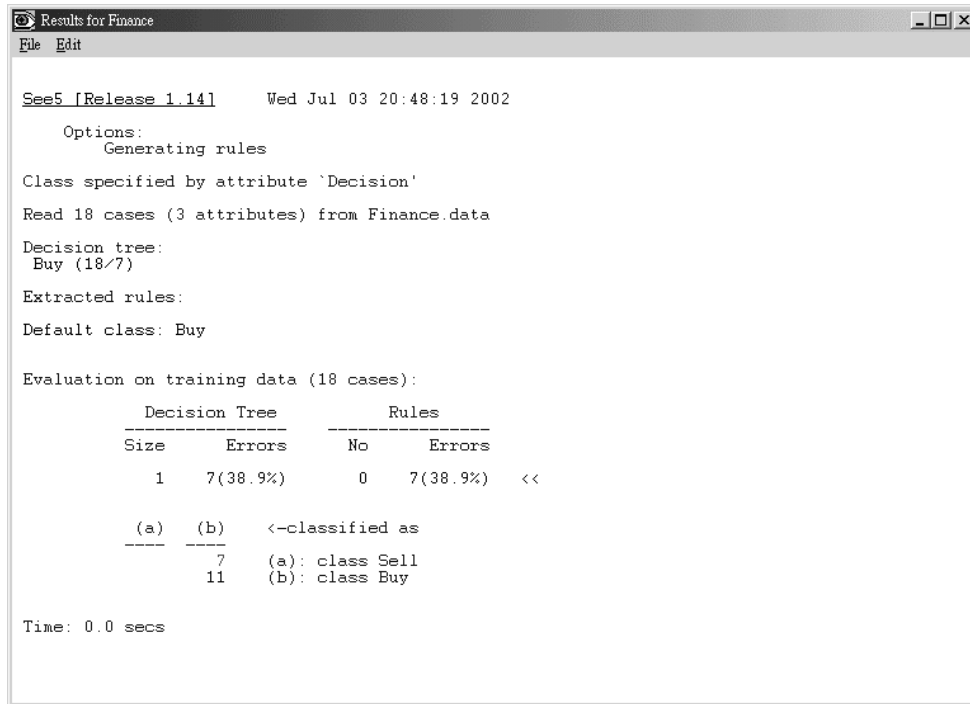


Figure 4 C5 Results Using Original Dataset

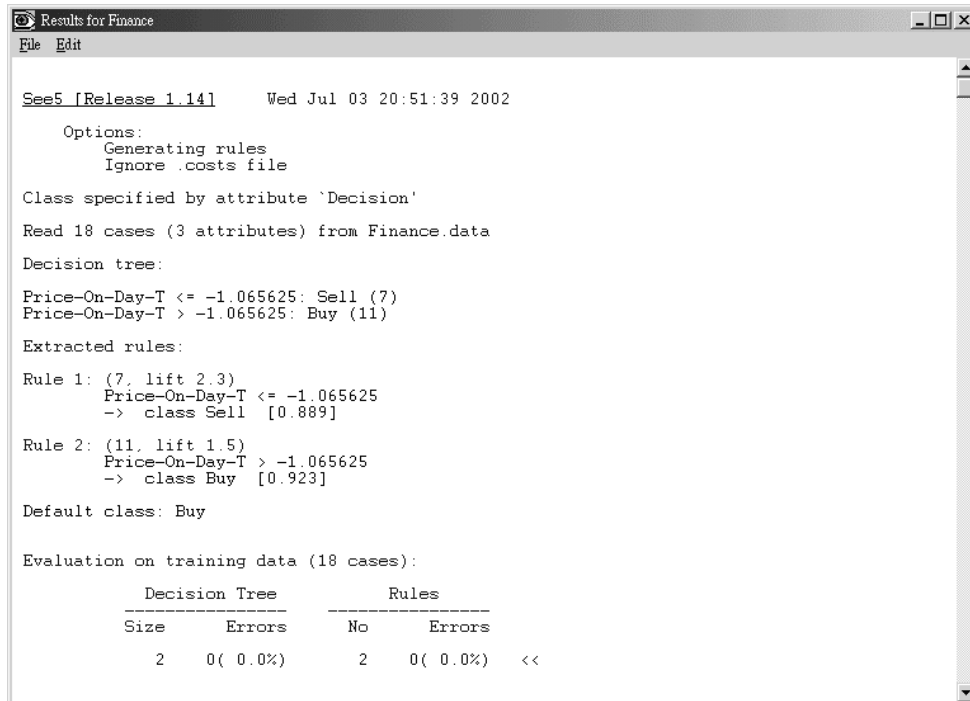


Figure 5 C5 Results Using Transformed Dataset (NDT Model)

From Figures 4 and 5, the comparisons are summarized as follows:

- After applying the NDT model, the error rate of classification for decision tree is reduced from 38.9 to 0.0.
- After applying the NDT model, the error rate of classification for decision rule is also reduced from 38.9 to 0.0.
- Before applying the NDT model, C5 generate no decision rules from the original dataset. This is because the original dataset contains noisy data.
- After applying the NDT model, C5 generate two decision rules from the transformed dataset. It reveals that the NDT model can handle the noisy data perfectly.

After removing the noisy data from original dataset, the original dataset contains 17 instances. Using this dataset, we also apply the NDT model to transform it. The results generated by the C5 are shown in Figures 6 and 7.

```

Results for Finance
File Edit
See5 [Release 1.14] Wed Jul 03 20:58:16 2002
Class specified by attribute 'Decision'
Read 17 cases (3 attributes) from Finance.data
Decision tree:
Price-on-Day-T+1 > 18.02: Buy (6)
Price-on-Day-T+1 <= 18.02:
... Price-On-Day-T > 17.84: Sell (4)
... Price-On-Day-T <= 17.84:
... Price-on-Day-T+1 <= 17.7: Sell (2)
... Price-on-Day-T+1 > 17.7: Buy (5)

Evaluation on training data (17 cases):
Decision Tree
-----
Size      Errors
4         0( 0.0%)  <<

(a)  (b)  <-classified as
-----
6         11  (a): class Sell
          (b): class Buy

Time: 0.0 secs
  
```

Figure 6 C5 Results Using Reduced Dataset

```

Results for Finance
File Edit

See5 [Release 1.14]   Wed Jul 03 21:03:33 2002

Options:
  Ignore .costs file

Class specified by attribute 'Decision'
Read 17 cases (3 attributes) from Finance.data

Decision tree:
Price-On-Day-T <= -1.065625: Sell (6)
Price-On-Day-T > -1.065625: Buy (11)

Evaluation on training data (17 cases):

      Decision Tree
      -----
      Size      Errors
      2         0( 0.0%)  <<

      (a)  (b)  <-classified as
      ----  ----
      6     11  (a): class Sell
                (b): class Buy

Time: 0.0 secs

```

Figure 7 C5 Results Using Transformed Reduced Dataset (NDT Model)

From Figures 6 and 7, the comparisons are summarized as follows:

- After applying the NDT model, the error rate of classification for decision tree and rule is still 0.0.
- After applying the NDT model, the decision tree size is reduced from 4 to 2.
- After applying the NDT model, the number of decision rules is reduced from 4 to 2.

From the above experiments, it is obviously that the NDT model remarkably improves the classification accuracy, decision tree size, and the number of decision rules based on C5. Besides, the NDT model can also perfectly handle the noisy data. From Figure 7, C5 generate the following rule.

If Price on Day T (in Table 3) ≤ -1.066 , Sell the Stock.

Otherwise, Buy the Stock.

“Price on Day T (in Table 3) ≤ -1.066 ” can be rewritten as follows:

Price on Day T (in Table 3) ≤ -1.066

$$\begin{aligned} &\equiv -38.387 * (\text{Price On Day T (in Table 1)} - 17.6) / 0.56 + 38.059 * \\ &\quad (\text{Price On Day T+1 (in Table 1)} - 17.6) / 0.56 - 0.099 > -1.066 \\ &\equiv \text{Price On Day T+1 (in Table 1)} + 0.15 > 1.009 * \text{Price On Day T} \\ &\Leftrightarrow \text{Price On Day T+1 (in Table 1)} > \text{Price On Day T (in Table 1)} \end{aligned}$$

Therefore, the correct rule “If Price on Day T+1 (in Table 1) > Price On Day T (in Table 1), Sell the Stock, Otherwise Buy the Stock” can be obtained. That is, the NDT model can actually discover the dependencies among attributes. In next section, we will demonstrate the NDT model in several larger datasets.

4 Experimental Results

In experiments, we use five datasets collected from the UCI database repository (<http://www1.ics.uci.edu/~mllearn/MLRepository.html>) as a test bed. Based on these datasets, we compare the NDT model with C5. Table 4 shows these five experimental datasets in details. They include four pure numerical datasets and 1 numerical-categorical-mixed dataset. The number of instances is listed in column named “#”.

Table 4 Experimental Datasets from UCI Databases

	Dataset Name	# of Input Attributes		Types in Target Class	#	Naive Prediction Error (%)
		Numerical	Categorical			
1	Wine	13	0	3	178	60.11%
2	Iris	4	0	3	150	66.67%
3	Pima	8	0	2	768	34.90%
4	Glass	9	0	6	214	64.49%
5	Heart	6	7	2	270	44.44%

Table 4 also lists the error rate for naive prediction, which is just classified the instances by the major proportion of target class. The naive prediction can be regarded as the baseline

model for the following experiments. That is, the accuracies of C5 and NDT model must far better than the naive prediction.

Table 5 Experimental Results for Decision Tree Using UCI Databases

	Dataset Name	#	C5		NDT		Naive Prediction Error (%)
			Decision Tree		Decision Tree		
			Size	Errors (%)	Size	Errors (%)	
1	Wine	178	5.40	7.35%	5.00	5.36%	60.11%
2	Iris	150	4.73	4.94%	3.94	3.41%	66.67%
3	Pima	768	26.65	25.77%	17.30	22.32%	34.90%
4	Glass	214	24.00	30.10%	21.69	29.52%	64.49%
5	Heart	270	21.62	22.96%	16.66	21.15%	44.44%

Table 6 Experimental Results for Decision Rule Using UCI Databases

	Dataset Name	#	C5		NDT		Naive Prediction Error (%)
			Decision Rule		Decision Rule		
			Size	Errors (%)	Size	Errors (%)	
1	Wine	178	4.85	6.95%	3.36	5.47%	60.11%
2	Iris	150	4.14	4.95%	3.03	3.48%	66.67%
3	Pima	768	17.12	25.50%	10.91	22.68%	34.90%
4	Glass	214	16.31	30.88%	15.94	29.60%	64.49%
5	Heart	270	11.31	21.27%	8.24	20.10%	44.44%

Table 7 Improvements of NDT Model Relative to C5 for Decision Tree

	Dataset Name	#	C5		NDT		Improvements (%)	
			Decision Tree		Decision Tree			
			Size	Errors (%)	Size	Errors (%)	Size	Errors (%)
1	Wine	178	5.40	7.35%	5.00	5.36%	7.41%	27.07%
2	Iris	150	4.73	4.94%	3.94	3.41%	16.70%	30.97%
3	Pima	768	26.65	25.77%	17.30	22.32%	35.08%	13.39%
4	Glass	214	24.00	30.10%	21.69	29.52%	9.64%	1.93%
5	Heart	270	21.62	22.96%	16.66	21.15%	22.94%	7.88%

Table 8 Improvements of NDT Model Relative to C5 for Decision Rule

	Dataset Name	#	C5		NDT		Improvements (%)	
			Decision Rule		Decision Rule			
			Size	Errors (%)	Size	Errors (%)	Size	Errors (%)
1	Wine	178	4.85	6.95%	3.36	5.47%	30.72%	21.29%
2	Iris	150	4.14	4.95%	3.03	3.48%	26.81%	29.70%
3	Pima	768	17.12	25.50%	10.91	22.68%	36.27%	11.06%
4	Glass	214	16.31	30.88%	15.94	29.60%	2.27%	4.15%
5	Heart	270	11.31	21.27%	8.24	20.10%	27.14%	5.50%

Tables 5 and 6 show the experimental results for decision tree and rule, respectively. Tables 7 and 8 show the improvements of NDT Model relative to C5 for decision tree and rule, respectively. From Tables 7 and 8, the improvements of NDT model relative to C5 are summarized as follows:

- After applying the NDT model, the reduction of decision tree size ranges from 7.41% to 35.08%.
- After applying the NDT model, the reduction of decision rule size ranges from 2.27% to 36.27%.
- After applying the NDT model, the reduction of the classification error rate for decision tree ranges from 1.93% to 30.97%.
- After applying the NDT model, the reduction of the classification error rate for decision rule ranges from 4.15% to 29.70%.

From the above experiments, it is obviously that the NDT model can actually discover the numerical-type dependencies among attributes. At the same time, the NDT model remarkably improves the classification accuracy, decision tree size, and the number of decision rules based on C5.

5 Concluding Remarks

In the past few years, many researchers focus on the research in classification and decision-tree learning algorithms. For those decision-tree learning algorithms, there are still challenging problems in real-life datasets, which are mixed with numerical and categorical attributes. In this paper, we propose a model, Neural Decision Tree (NDT) Model, to deal with the problems of attribute dependencies. It combines the neural network technologies and traditional decision tree learning capabilities to handle the complicated and real cases. In experiments, we use five real datasets from the UCI databases. Based on these datasets, the experimental results show that the NDT model remarkably improves the classification accuracy, decision tree size, and the number of decision rules based on C5. This is because the NDT model can successfully capture the dependencies among attributes.

Even though the NDT model performs well for improving the accuracy and effectiveness of C5, the dependencies among categorical-type attributes are not processed in this paper. Chen [8] proposed the concepts for categorical-type dependencies. However, they did not describe how these dependencies could be obtained efficiently. This kind of dependencies is also important for the improvements of decision-tree learning algorithms. If we can capture the complete dependencies among input attributes, more improvements can be reached. But, it needs to be investigated further.

Acknowledgement

Research on this paper was partially supported by National Science Council grant NSC90-2213-E130-003 and NSC90-2213-E030-003.

References

- [1] M. S. Chen, J. Han and S. Yu, (1996). “Data Mining: An Overview from a Database Perspective”, *IEEE Transaction on Knowledge and Data Engineering*, Vol. 8, No. 6, pp. 866-882, 1996.
- [2] J. R. Quinlan, (1996). “Improved Use of Continuous Attributes in C4.5”, *Journal of Artificial Intelligence Approach*, No. 4, pp. 77-90, 1996.
- [3] B. Kovalerchuk and E. Vityaev, (2000). *Data Mining In Finance - Advances in Relational and Hybrid Methods*, Kluwer Academic Publishers, 2000.
- [4] H. Lu, R. Setiono and H. Liu, (1996) “Effective Data Mining Using Neural Networks”, *IEEE Transaction on Knowledge and Data Engineering*, Vol. 8, No. 6, pp. 957-961, 1996.
- [5] S. W. Changchien and T. C. Lu, (2000). “A Data Mining Procedure Using Neural Network, Self Organization Map and Rough Set to Discover Association Rules”, *Proceedings of the International Computer Symposium*, 2000.
- [6] J. Han and M. Kamber, (2000). *Data Mining - Concepts and Techniques*, Morgan Kaufmann Publishers, 2000.
- [7] D. Pyle, (1999). *Data Preparation for Data Mining*, Morgan Kaufmann Publishers, 1999.
- [8] M. S. Chen, (1998). “On the Evaluation of Using Multiple Attributes for Mining Classification Rules”, *Proceedings of IEEE International Conference On Tools and Artificial Intelligence*, 1998.