International Journal of Computer Theory and Engineering, Vol. 2, No. 4, August, 2010
1793-8201

Extracting Prediction Rules for Loan Default Using Neural Networks through Attribute Relevance Analysis

M. V. Jagannatha Reddy and Dr. B.Kavitha

***Abstract*—Predicting the class label loan defaulter using neural networks through attribute relevance analysis is presented in the previous paper. In this paper we are extracting prediction rules from the predicted class label. This method has the advantage that the number of units required can be reduced using attribute relevance analysis. So that we can increase the speed of neural network technique for predicting the class label of the new tuples. In this proposed paper attribute relevance analysis is used to eliminate irrelevant attributes to give as inputs to neural network. Neural networks used for predicting the class label and deriving the prediction rules from the class label. These rules are more useful in understanding the customer.**

***Index Terms*—Classification rules, Attribute relevance analysis, neural networks, prediction rules, defaulter, class label.**

I. Introduction

A. Data Mining:

Data mining is the non-trivial process of identifying valid, novel, potentially useful and ultimately understandable patterns in data [3]. The term data mining refers to finding of relevant and useful information from databases. Data Mining and knowledge discovery in database is a new interdisciplinary field, integrating ideas from statistics, machine learning, databases and parallel computing. We can perform data mining on different kinds of databases. The kind of patterns can be mined form these databases are characterization and comparison, association rules, classification and prediction, cluster analysis, evolution analysis etc.,

B. Classification and Prediction

Databases contain many hidden information that can be used by making an intelligent decision. The two forms of data analysis are Classification and Prediction. It is used to extract describing important data classes or to predict future data trends. Classification identifies data into a predefined groups or classes. Before examining the data, the classes are determined because it is supervised learning. For instance, a bank loan officer classifies the application are analyzed and determined whether to make a bank loan and identifying the credit risk. In these examples, the classification is the data analysis. Where the models or classifiers is established to predict the categorical labels such as "yes" or "no" to give a loan.

Prediction is a method which is used to estimate the future data based on past and current data. In many realistic data mining applications, it can be seen prediction is a type of classification. The difference between the classification and prediction is classification determines the present aspect where as prediction estimates the future aspect. The patterns extracted from this technique is useful for decision making.

Prediction can be viewed as the construction and the use of a model to asses the class of an unlabelled sample.

Data prediction is a two step process.

1) Prediction does not have "class label attribute" because the attribute for which values are being predicted is a continuous valued (ordered) rather than categorical
2) The accuracy of a predictor is estimated by computing an error based on the difference between the predicted value and the actual known value for each test tuple.

The major issues in classification and prediction is

1) Preparing the data for classification and prediction.
2) Comparing the classification and prediction methods.

For the first issue, the following preprocessing steps may be applied to the data which helps to improve the accuracy, efficiency and scalability.

- **Data Cleaning:** To remove or reduce noise and missing values.
- **Relevance analysis:** relevance analysis removes redundant and irrelevant data from the task relevant data.
- **Data transformation and reduction:** The data may be transformed by normalization. Normalization involves scaling all values for a given attribute so that they fall with in a small specified range such as from 0 to 1.
- **Generalization:** It compresses the original training data, fewer input/output operations may be involved during learning.

For the second issue, Classification and prediction can be compared and evaluated according to the following criteria

- **Accuracy:** The accuracy of a classifier refers to ability of a given classifier to correctly predict the class label of new data.
- **Speed:** This involves the computational cost.
- **Robustness:** This is the ability of classifier or predictor to make correct prediction given noisy or data with missing values.
- **Scalability:** This refers to ability to construct the classifier or predictor given large amounts of data.
- **Interpretability:** This refers to the level of understanding and insight that is provided by the classifier or predictor.

In this paper neural networks are used as classifier for predicting the class label defaulter.

C. Prediction using neural networks:

[5]The advantage of the usage of neural networks for prediction is that they are able to learn from examples only and that after their learning is finished, they are able to catch hidden and strongly non-linear dependencies, even when there is a significant noise in the training set.

D. Neural network:

Neural network is a set of connected input/output units where each connection has a weight associated with it. During the learning phase, the network learns by adjusting the weights so as to be able to predict the correct class label of the input samples [4].

[1]It is difficult to say exactly when the first neural network on the computer was built. During the world war II a seminal paper was published by McCulloch and Pitts which first outlined the idea that simple processing units (like the individual neurons in the human brain) could be connected together in large networks to create a system that could solve difficult problems and display behavior that was much more complex than the simple pieces that made it up. Since that time much progress has been made in finding ways to apply artificial neural networks to real world prediction problems and in improving the performance of the algorithm in general. In many respects the greatest breakthroughs in neural networks in recent years have been in their application to more common real world problems like customer response prediction or fraud detection rather than the loftier goals that were originally set out for techniques such as overall human learning and computer speech and image understanding

II. Prediction Rules

A. What is a rule?

In rule induction systems the rule itself is of a simple form of "if this and this and this then this". For example a rule that a supermarket might find in their data collected from scanners would be: "if pickles are purchased then ketchup is purchased'. Or

- If paper plates then plastic forks
- If dip then potato chips
- If salsa then tortilla chips

In order for the rules to be useful there are two pieces of information that must be supplied as well as the actual rule:

- Accuracy - How often is the rule correct?
- Coverage - How often does the rule apply?

Just because the pattern in the data base is expressed as rule does not mean that it is true all the time. Thus just like in other data mining algorithms it is important to recognize and make explicit the uncertainty in the rule. This is what the accuracy of the rule means

B. What to do with a rule

When the rules are mined out of the database the rules can be used either for understanding better the business problems that the data reflects or for performing actual predictions against some predefined prediction target. Since there is both a left side and a right side to a rule (antecedent and consequent) they can be used in several ways for your business.

Target the antecedent. In this case all rules that have a certain value for the antecedent are gathered and displayed to the user. For instance a grocery store may request all rules that have nails, bolts or screws as the antecedent in order to try to understand whether discontinuing the sale of these low margin items will have any effect on other higher margin. For instance maybe people who buy nails also buy expensive hammers but wouldn't do so at the store if the nails were not available.

Target the consequent. In this case all rules that have a certain value for the consequent can be used to understand what is associated with the consequent and perhaps what affects the consequent. For instance it might be useful to know all of the interesting rules that have "coffee" in their consequent. These may well be the rules that affect the purchases of coffee and that a store owner may want to put close to the coffee in order to increase the sale of both items. Or it might be the rule that the coffee manufacturer uses to determine in which magazine to place their next coupons.

Target based on accuracy. Some times the most important thing for a user is the accuracy of the rules that are being generated. Highly accurate rules of 80% or 90% imply strong relationships that can be exploited even if they have low coverage of the database and only occur a limited number of times. For instance a rule that only has 0.1% coverage but 95% can only be applied one time out of one thousand but it will very likely be correct. If this one time is highly profitable that it can be worthwhile. This, for instance, is how some of the most successful data mining applications work in the financial markets - looking for that limited amount of time where a very confident prediction can be made.

Target based on coverage. Some times user want to know what the most ubiquitous rules are or those rules that are most readily applicable. By looking at rules ranked by coverage they can quickly get a high level view of what is happening within their database most of the time.

Target based on "interestingness". Rules are interesting when they have high coverage and high accuracy and deviate from the norm. There have been many ways that rules have been ranked by some measure of interestingness so that the trade off between coverage and accuracy can be made.

C. Prediction

After the rules are created and their interestingness is measured there is also a call for performing prediction with the rules. Each rule by itself can perform prediction - the consequent is the target and the accuracy of the rule is the accuracy of the prediction. But because rule induction systems produce many rules for a given antecedent or consequent there can be conflicting predictions with different accuracies. This is an opportunity for improving

the overall performance of the systems by combining the rules. This can be done in a variety of ways by summing the accuracies as if they were weights or just by taking the prediction of the rule with the maximum accuracy.

D. The General Idea:

The general idea of a rule prediction system is that rules are created that show the relationship between events captured in your database. These rules can be simple with just one element in the antecedent or they might be more complicated with many column value pairs in the antecedent all joined together by a conjunction (item1 and item2 and item3 … must all occur for the antecedent to be true).

The rules are used to find interesting patterns in the database but they are also used at times for prediction. There are two main things that are important to understanding a rule:

Accuracy - Accuracy refers to the probability that if the antecedent is true that the precedent will be true. High accuracy means that this is a rule that is highly dependable.

Coverage - Coverage refers to the number of records in the database that the rule applies to. High coverage means that the rule can be used very often and also that it is less likely to be a spurious artifact of the sampling technique or idiosyncrasies of the database

E. The business importance of accuracy and coverage

From a business perspective accurate rules are important because they imply that there is useful predictive information in the database that can be exploited - namely that there is something far from independent between the antecedent and the consequent. The lower the accuracy the closer the rule comes to just random guessing. If the accuracy is significantly below that of what would be expected from random guessing then the negation of the antecedent may well in fact be useful (for instance people who buy denture adhesive are much less likely to buy fresh corn on the cob than normal).

From a business perspective coverage implies how often you can use a useful rule. For instance you may have a rule that is 100% accurate but is only applicable in 1 out of every 100,000 shopping baskets. You can rearrange your shelf space to take advantage of this fact but it will not make you much money since the event is not very likely to happen. Table 1. Displays the trade off between coverage and accuracy.

TABLE.1 TRADE OFF BETWEEN COVERAGE AND ACCURACY

Accuracy Low	**Accuracy High**
Rule is rarely correct but can be used often.	Rule is often correct and can be used often.
Rule is rarely correct and can be only rarely used.	Rule is often correct but can be only rarely used.

F. Interestingness:

- Interestingness increases as accuracy increases (or decreases with decreasing accuracy) if the coverage is fixed.
- Interestingness increases or decreases with coverage if accuracy stays fixed
- Interestingness decreases with coverage for a fixed number of correct responses (remember accuracy equals the number of correct responses divided by the coverage).

III. ATTRIBUTE RELAVENCE ANALYSIS

Task relevant data is selected from the relational data base or data warehouse query processing. On this data attribute relevance analysis is performed..

Attribute relevance analysis [2] is performed in order to eliminate irrelevant or weakly relevant or less informative attributes, and retain the most relevant attributes for analysis. This reduces number of units required for the neural network model. As a result the complexity of constructing network topology makes easy. Also since number of units reduces its speed increases. To perform attribute relevance analysis information gain measure is used.

A. Information Gain measure:

In order to find the gain of each attribute. Let S be a set of training samples, where the class label of each sample is known. Each sample is in fact a tuple. Suppose that there are m classes $c_1, c_2 \ldots c_m$. The expected information needed to classify a given sample is

$$I(s_1, s_2, \ldots, s_m) = -\sum_{i=1}^{m} \frac{s_i}{s} \log_2 \frac{s_i}{s} \quad (1)$$

Where S_i is the total number of samples that belongs to class C_i and S is total number of samples.

An attribute A with values can be used to partition S in to subsets $\{s_s, s_2, \ldots . s_v\}$. The expected information based on this partitioning is known as entropy of A. it is calculated as

$$E(A) = \sum_{j=1}^{v} \frac{s_{1j} + \ldots + s_{mj}}{s} I(s_{1j}, \ldots, s_{mj}) \quad (2)$$

The information gain obtained by this partitioning on A is defined by

$$Gain(A) = I(s_1, s_2, \ldots, s_m) - E(A) \quad (3)$$

In this approach to relevance analysis, we can compute the information gain for each of the attributes. The attributes with the highest information gain is considered as most important of the given set. We can eliminate irrelevant attributes by using threshold levels.

In our discussion it is found that from the given training samples whose class labels are known and using the above equations Age and Income are the most relevant attributes that are found from given threshold value and they are used for prediction of loan defaulter.

B. Attribute Generalization:

Based on the examination of number of distinct values of each attribute in the relevant set of data. The generalization is performed by either attribute removal or attribute generalization. Aggregation is performed by merging identical generalized tuples.

- *Attribute removal:* is based on the following rule: if there is a large set of distinct values for an attribute but either (1) there is no generalization

operator as the attribute or (20 its higher level concepts are expressed in terms of other attributes, then the attribute should be removed from the working relation.

- *Attribute generalization:* Is based on the following rule: If there is a large set of distinct values for an attribute in the relation and there exists a set of generalization operators the generalization operator should be selected and applied to the attribute.

IV. Designing A Neural Network

A neural network is loosely based on how some people believe that the human brain is organized and how it learns. Given that there are two main structures of consequence in the neural network:

The node - which loosely corresponds to the neuron in the human brain. Where input values are applied to the node.

The link - This loosely corresponds to the connections between neurons (axons, dendrites and synapses) in the human brain. The connecting links provide the weights.

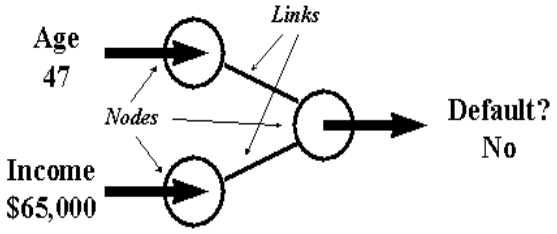

Figure 1. A simplified view of a neural network for prediction of loan default.

In Figure 1 there is a drawing of a simple neural network [1]. The round circles represent the nodes and the connecting lines represent the links. The neural network functions by accepting predictor values at the left and performing calculations on those values to produce new values in the node at the far right. The value at this node represents the prediction from the neural network model. In this case the network takes in values for predictors for age and income and predicts whether the person will default on a bank loan.

The neural network model is created by presenting it with many examples of the predictor values from records in the training set (in this example age and income are used) and the prediction value from those same records. By comparing the correct answer obtained from the training record and the predicted answer from the neural network it is possible to slowly change the behavior of the neural network by changing the values of the link weights. In some ways this is like having a grade school teacher asks questions of her student (a.k.a. the neural network) and if the answer is wrong to verbally correct the student. The greater the error the harsher the verbal correction. So that large errors are given greater attention at correction than are small errors.

For the actual neural network it is the weights of the links that actually control the prediction value for a given record. Thus the particular model that is being found by the neural network is in fact fully determined by the weights and the architectural structure of the network. For this reason it is the link weights that are modified each time an error is made.

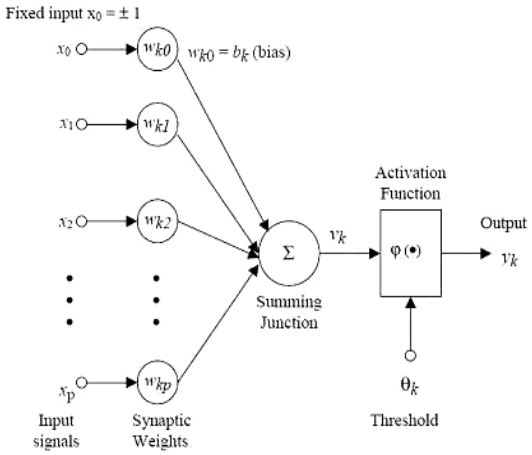

Figure 2. Node

Neural Network is the mathematical model of a neuron (node) as shown in Figure (2). The three basic components of the neuron are:

(1) Connecting links that provide weights

(2) An adder that sums the weighted input values to compute the input to the activation function

(3) Activation function maps a large input domain into a smaller range of 0 to 1

Network design is a trial and error process and may affect the accuracy of the resulting trained network. Then initial values of the weights may also affect the resulting accuracy. Once the network has been trained and its accuracy is not acceptable, it is common to repeat the training process with different network topology or different set of weights. Cross validation techniques for accuracy estimation can be used to help to decide when an acceptable network has been found.

After designing a classifier, neural network highly relevant attributes are applied to the input nodes. These attribute values are then multiplied with weights and all these products are summed at the output of adder. This value gives the class label.

V. The Learning That Goas on in the Hidden Nodes

The learning procedure for the neural network has been defined to work for the weights in the links connecting the hidden layer. A good metaphor for how this works is to think of a military operation in some war where there are many layers of command with a general ultimately responsible for making the decisions on where to advance and where to retreat. The general probably has several lieutenant generals advising him and each lieutenant general probably has several major generals advising him. This hierarchy continuing downward through colonels and privates at the bottom of the hierarchy.

This is not too far from the structure of a neural network with several hidden layers and one output node. You can think of the inputs coming from the hidden nodes as advice. The link weight corresponds to the trust that the general has in his advisors. Some trusted advisors have

very high weights and some advisors may not be trusted and in fact have negative weights. The other part of the advice from the advisors has to do with how competent the particular advisor is for a given situation. The general may have a trusted advisor but if that advisor has no expertise in aerial invasion and the question at hand has to do with a situation involving the air force this advisor may be very well trusted but the advisor himself may not have any strong opinion one way or another.

In this analogy the link weight of a neural network to an output unit is like the trust or confidence that a commander has in his advisors and the actual node value represents how strong an opinion this particular advisor has about this particular situation. To make a decision the general considers how trustworthy and valuable the advice is and how knowledgeable and confident each advisor is in making their suggestion and then taking all of this into account the general makes the decision to advance or retreat.

In the same way the output node will make a decision (a prediction) by taking into account all of the input from its advisors (the nodes connected to it). In the case of the neural network this decision is reach by multiplying the link weight by the output value of the node and summing these values across all nodes. If the prediction is incorrect the nodes that had the most influence on making the decision have their weights modified so that the wrong prediction is less likely to be made the next time.

This learning in the neural network is very similar to what happens when the wrong decision is made by the general. The confidence that the general has in all of those advisors that gave the wrong recommendation is decreased - and all the more so for those advisors who were very confident and vocal in their recommendation. On the other hand any advisors who were making the correct recommendation but whose input was not taken as seriously would be taken more seriously the next time. Likewise any advisor that was reprimanded for giving the wrong advice to the general would then go back to his advisors and determine which of them he had trusted more than he should have in making his recommendation and who he should have listened more closely to.

VI. Implementation

In order to make a prediction the neural network accepts the values for the predictors on what are called the input nodes. These become the values for those nodes, those values are then multiplied by values that are stored in the links (sometimes called links and in some ways similar to the weights that were applied to predictors in the nearest neighbor method). These values are then added together at the node at the far right (the output node) a special threshold function is applied and the resulting number is the prediction. In this case if the resulting number is 0 the record is considered to be a good credit risk (no default) if the number is 1 the record is considered to be a bad credit risk (likely default).

Here the attribute values retained by attribute relevance analysis are applied as inputs to neural network. Simplified version of the calculations made in Figure 1 might look like what is shown in Figure 3. Here the value age of 47 is normalized to fall between 0.0 and 1.0 and has the value 0.47 and the income is normalized to the value 0.65. This simplified neural network makes the prediction of no default for a 47 year old making $65,000. The links are weighted at 0.7 and 0.1 and the resulting value after multiplying the node values by the link weights is 0.39. The network has been trained to learn that an output value of 1.0 indicates default and that 0.0 indicate non-default. The output value calculated here (0.39) is closer to 0.0 than to 1.0 so the record is assigned a non-default

The normalized input values are multiplied by the link weights and added together at the output as shown in figure.3. The weights are adjusted by calculating the error between predicted class label and actual class label by using training set. This weights adjustment is made using back propagation technique.

A very similar method of training takes place in the neural network. It is called "back propagation" and refers to the propagation of the error backwards from the output nodes (where the error is easy to determine the difference between the actual prediction value from the training database and the prediction from the neural network) through the hidden layers and to the input layers. At each level the link weights between the layers are updated so as to decrease the chance of making the same mistake again.

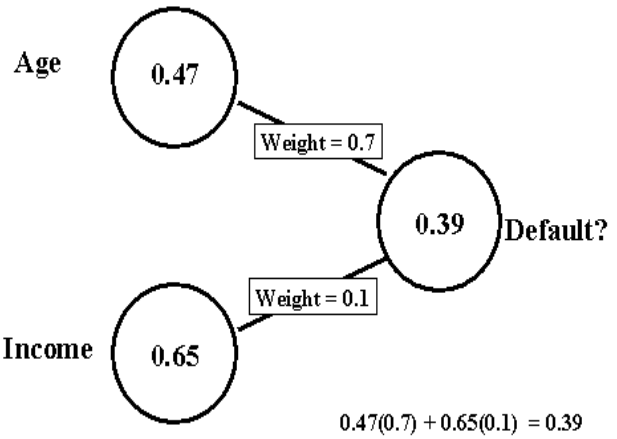

Figure 3.

A. Assumptions

If age is in between 21—30 then weight is 0.1

If age is in between 31—40 then weight is 0.4
If age is in between 41—50 then weight is 0.7
If age is in between 51—60 then weight is 0.8
If age is greater than 60 then weight is 0.9
If income is between 21k—30k then wt is 0.9
If income is between 31k—40k then wt is 0.7
If income is between 41k—50k then wt is 0.5
If income is between 51k—60k then wt is 0.3
If income is greater than 61k then wt is 0.1

B. Calculations

Output = age (wt) + income (wt)

TABLE. 2 SAMPLE CALCULATIONS TABULATED

Sl.no	Age (yr)	Income (k)	Output	Default
1	47	65	0.394	No
2	32	42	0.338	No
3	50	40	0.630	Yes
4	60	25	0.705	Yes
5	21	62	0.083	No
6	45	55	0.480	No
7	55	70	0.510	Yes
8	28	35	0.273	No
9	45	45	0.540	Yes
10	60	50	0.730	Yes

VII. DERIVED PREDICTION RULES

IF Age =47 and Income = $65K THEN Loan = No default

IF Age =32 and Income =$42K THEN Loan = No default

IF Age =21 and Income = $62K THEN Loan = No default

IF Age =45 and Income = $55K THEN Loan = No default

IF Age =28 and Income = $35K THEN Loan = No default

IF Age =50 and Income = $40K THEN Loan = default

IF Age =60 and Income = $25K THEN Loan = default

IF Age =55 and Income = $70K THEN Loan = default

IF Age =45 and Income = $45K THEN Loan = default

IF Age =60 and Income = $50K THEN Loan = default

VIII. CONCLUSION AND FUTURE WORK

This neural network model is implemented and prediction rules are extracted from the output of neural networks and using the input values. It is tested for ten rules as shown above. The accuracy is appreciable and we can still improve the accuracy by calculating the error in wrong predicted rules by adjusting the weights of the neural networks. Since we used attribute relevance analysis the attributes retained for predicting the class label is very less. Hence the number of input nodes required in constructing the neural network model is also less. This shows that complexity in choosing the nodes reduces. Neural networks for predicting the rules is feasible. In the future work we can use the neural networks for predicting the association rules.

REFERENCES

[1] http://www.amazon.com
[2] An Overview of Data Mining Techniques. by Alex Berson, Stephen Smith, and Kurt Thearling
[3] Jiawei Han Micheline Kamber, Data Mining Concepts and Techniques.
[4] Arun K Pujari, Data Mining Techniques, University press. India www.google.com
[5] http://www.obitko.com/tutorials/neural-network-prediction/prediction-using-neural-networks.html