CRNN: Integrating Classification Rule and Neural Network

Wei Li, Longbing Cao and Dazhe Zhao

***Abstract*— Association classification has been an important type of the rule based classification. A variety of approaches have been proposed to build a classifier based on classification rules. In the prediction stage of the extant approaches, most of the existing association classifiers use the ensemble quality measurement of each rule in a subset rules to predict the class label of the new data. This method still suffers the following two problems. The classification rules are used individually that the coupling relations between rules are ignored in the prediction. However, in real-word rules set, the rules are often inter related together and many rules are usually satisfied partially when a new data object comes. Furthermore, the classification rule based prediction model lacks a general expression of the decision methodology. This paper proposes a classification method that integrating classification rule and neural network (CRNN for short), which gives a general form of the rule based decision methodology by rules network. In comparison with the extant rule based classifiers, such as C4.5, CBA, CMAR and CPAR, our approach has two advantages. First, CRNN takes into account of the coupling relations between rules from the training data in the prediction step. Second, CRNN obtains higher performance on the structure and parameter learning automatically than traditional neural network. CRNN uses the linear computing algorithm in neural network instead of the costly iterative learning algorithm. Two ways of the classification rule set generation are conducted in this paper for the CRNN evaluation and CRNN achieves the satisfied performance.**

I. Introduction

IN recent years, the association rule mining integrated with classification, which is called association classification, has been widely studied [1][3]. The performance of association classification, such as CBA [2][4] and CMAR [5], shows that it is even better than traditional rule based classifier such as C4.5 [6]. These methods use association rule mining algorithm, such as Apriori [7], FP-growth [8], to generate a lot of rules and adopt strategies to select useful rules for the classification tasks. The general prediction schema of the rule based classification is shown as in Figure 1. Three methods are usually used in the prediction task, which are single rule based prediction, *top-K* rules based prediction and group rules together based prediction.

As shown in Fig. 1, the single rule based prediction relies on the sorting of the rules, and the first satisfied rules class is thought as the predicting result. It is more reliable using *top-K* satisfied rules to make the final decision in the second approach. The measurement (e.g. expected accuracy is used in [14] of each rule is conducted, and then makes an average value on the K rules that belong to same class. Accordingly, the class with the highest value is selected. However, the rules are not always consistent in class tags for a new coming data object. Therefore, the rules are grouped by the class labels, and then the classifier uses the overall effects of the group rules and yields to the group with highest total measurement, which is as shown in the third method. E.g. χ^2 is used to qualify every rule in the group [5], and then a total sum (or weighted sum) value is obtained. Finally, the label of the group with the highest χ^2 value is chosen as the predicting outcome.

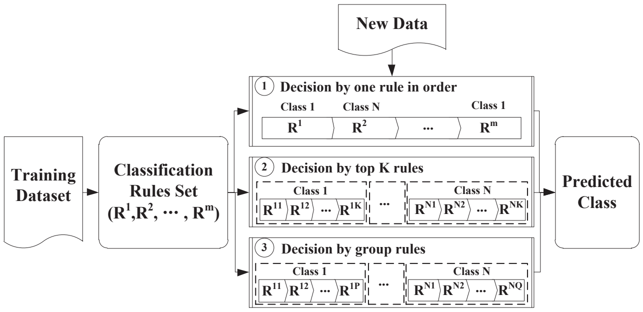

New Data
Training Dataset
Classification Rules Set (R^1,R^2, ⋯ , R^m)
1 Decision by one rule in order
Class 1
Class N
Class 1
R^1
R^2
...
R^m
2 Decision by top K rules
Class 1
R^{11} R^{12} ... R^{1K}
...
Class N
R^{N1} R^{N2} ... R^{NK}
3 Decision by group rules
Class 1
R^{11} R^{12} ... R^{1P}
...
Class N
R^{N1} R^{N2} ... R^{NQ}
Predicted Class

Fig. 1. The classification rule based prediction methodology

However, the methods shown in Fig. 1 may suffer some weakness as shown below.

1) *The coupling relations between the rules are ignored in prediction stage*. The rules are treated individually facing the new data object. In fact, the rules bodies often contain same items to each other. The new data object may be fully or partly satisfy to many rules, and these rules have similar rule bodies that means the rules have inter relation and synthesize effect on the class prediction. Moreover, a subset rules are used in the prediction decision as shown in Fig. 1 in extant approaches.
2) *A general expression of the rule based decision methodology is lacked at present.* The rule's support and confidence, which are originally used to measure the rules, are abandoned in these methods but using some alternative measurements which have the similar functions as support and confidence theoretically. For example, the Laplace accuracy of a rule r is defined as $(n_c+1)/(n_{tot}+k)$, where k is the number of classes, n_{tot} is the total number of data objects that satisfy the rule body p, and n_c data objects belong to class c. However, the value of Laplace accuracy is just the

Wei Li and Longbing Cao are with the Department of Advanced Analytics Institute University of Technology, Sydney, Australia(email: lewe01@gmail.com, longbing.cao@uts.edu.au).

Dazhe Zhao is with the Department of Key Laboratory of Medical Image Computing of Ministry of Education Northeastern University, Shenyang, China (email: zhaodz@neusoft.com).

This work is in part by Australian Research Council Discovery Grants (DP1096218 and DP130102691) and ARC Linkage Grant (LP100200774), and by Chinese NSF grants (61172002 and 61001047) and the Central University project N110618001.

confidence of rule when the data set size N is big enough.

Proof: $(n_c + 1)/(n_{tot} + k) = (n_c/N + 1/N)/(n_{tot}/N + k/N) \approx (n_c/N)/(n_{tot}/N) = support(r)/support(p) = confidence(r)$, where $1/N \approx 0$ and $k/N \approx 0$. ■

In order to solve the above two problems, the rules are used in prediction model as rules network. The support and confidence are also integrated into the classification model. The idea of this paper is from artificial neural network [10] (ANN for short) which can cover all the rules and rules parameters. As we known, the ANN classifier has been studied in machine learning for many years and it has been used in a variety of applications. ANN has many advantages in practice, but a coin has two sides. The training process is usually longer than other classification approaches; moreover, the structure of ANN needs to be design manually and the hidden nodes form black box hard for users understanding.

This paper proposes a new neural network structure based on the classification rules (CRNN for short), which integrates the advantages of association classification and neural networks and is different from [9]. This new classification method makes full use of the rules and obtains much more efficient in both the network structure and parameters learning than the traditional neural network. In our approach, CRNN relies on the rules mined from the training data set, so the quality of rules has still impact on the performance of CRNN. We use two different rules set generation methods, which are the association rule mining with rule selection strategy [5][8] and FOIL (First Order Inductive Learner) based rule searching with the rule support and confidence [11], to conduct the CRNN prediction accuracy evaluation. CRNN tackles the problems of the general expression of rule based prediction decision and complex computing process of the structure and parameter learning in neural network. The contributions of this study are as follows.

1) We propose a method to integrate all the classification rules into classification model as well as the coupling relations between rules.
2) This paper introduces a new structure of neural network. CRNN is created by the rules set and the structure of CRNN is determined automatically.
3) CRNN shows a new approach to obtain the parameters in neural network efficiently. The parameters in CRNN are fast obtained once the classification rule is given.

The subsequent chapters are organized as follows. Section II describes the definition of the research problem and an overview of the proposed method. Section III introduces the two methods of classification rule generation. Section IV shows the design and construction of the CRNN model. CRNN classification is presented in Section V using the CRNN model. Our approach is evaluated in Section VI and we make the conclusions in Section VII.

II. Problem Statement

Given a set of data items $I = I_1, I_2, \cdots, I_n$, where $I_1, I_2, \cdots, I_n$ are items of attributes values. Let D denote the data set which has N data records. Each record contains a number of attributes and every attribute contains a subset items in I. Each continuous attribute should be discretized firstly into categorical attribute items. Each data record corresponds to a class tag of Y, where $Y = \{Y_1, ..., Y_M\}$.

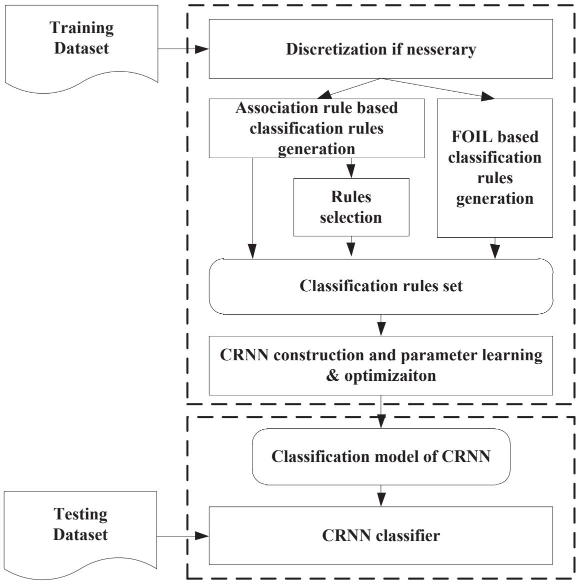
Definition 1: (Classification Rule) Let pattern A contain one or more data items combined as $I_i^{\wedge} \cdots {}^{\wedge} I_j$, where A links to a class tag Y_k, we call the form, $r : I_i^{\wedge} \cdots {}^{\wedge} I_j \rightarrow Y_k$, a *Classification Rule*. The support of the classification rule r is defined as the percentage of the pattern A in the data set, written as $sup(r)$.

$$sup(r) = sup(A^{\wedge}Y_k) = \frac{\#(A^{\wedge}Y_k)}{N} \tag{1}$$

where $\#(A^{\wedge}Y_k)$ is the size of the data records that cover pattern A and belong to class Y_k . The confidence of the classification rule, $conf(r)$, is defined as following.

$$conf(r) = \frac{sup(A^{\wedge}Y_k)}{sup(A)} = \frac{sup(r)}{sup(A)} \tag{2}$$

The problem we want to solve is as following. Defining a model F based on neural network that incorporates the classification rule set, $\{r_1, r_2, ..., r_p\}$, from data set D, then making prediction for the new data record, x ,using the model F, be $Y_k = F(r_1, r_2, ..., r_p)|_x$.



Training Dataset
Discretization if nesserary
Association rule based classification rules generation
FOIL based classification rules generation
Rules selection
Classification rules set
CRNN construction and parameter learning & optimizaiton
Classification model of CRNN
Testing Dataset
CRNN classifier

Fig. 2. CRNN framework

The framework of the CRNN is shown as Fig. 2. The approach is divided into two parts. Firstly, two ways are shown to generate the classification rules sets. CRNN model is design theoretically and then the construction algorithm is introduced. Secondly, the classification procedure of the CRNN is described using the above model. 21 data sets are used to evaluate the classifier and the approach achieves satisfied performance.

III. Classification Rule Generation

In this paper, two different ways are used to generate the rules set, that are the association rule mining (ARM for short) based rule generation and the FOIL based rule generation. A small data set is shown in Example 1.

Example 1 (Mining Classification Rules) Let D is the training data set as Table I (the first 6 columns). There are four attributes in every record and three class labels in total.

TABLE I
Data set for example 1

TID	A	B	C	D	Class	Transaction record
1	a_2	b_1	c_2	d_1	Y_1	a_2, b_1, c_2, d_1, Y_1
2	a_1	b_2	c_1	d_2	Y_2	a_1, b_2, c_1, d_2, Y_2
3	a_2	b_3	c_2	d_3	Y_1	a_2, b_3, c_2, d_3, Y_1
4	a_1	b_2	c_3	d_3	Y_3	a_1, b_2, c_3, d_3, Y_3
5	a_1	b_2	c_1	d_3	Y_3	a_1, b_2, c_1, d_3, Y_3
6	a_1	b_3	c_1	d_1	Y_2	a_1, b_3, c_1, d_1, Y_2
7	a_1	b_3	c_3	d_2	Y_1	a_1, b_3, c_3, d_2, Y_1

A. ARM Based Rule Generation

A new data set D' is organized based on the original data set D, in which each record is converted to a transaction data record with the class label Y_i. The new data set is shown as Table I (the last column). This new data set is easy to be processed by the traditional association rule mining methods. The process of ARM based rule generation is different from traditional association rule mining that the rule needs consist of a class tag Yi as right part of the rule. The ARM based classification rule generation is defined as in Algorithm 1.

```
Algorithm 1: ARM based rule generation. ARM-rg
Data: A transaction database, $D$; the minimum support
    threshold, $\sigma$.
Result: A rule set, $RuleSet$.
begin
    initial $RS = \emptyset$
    find all the frequent patternset, $pts$, that meet support $\sigma$
    initial pattern and support map table, $patternset = \{\}$
    for each $p$ in $pts$ do
        given pattern $p = (A \wedge Y_i, sup)$ or $p = (A, sup)$ if $p$
        doesnt contain class label $Y_i$ then
            add $p$ in to $patternset$, $patternset[A] = sup$
    for each $p$ in $pts$ do
        given pattern $p = (A \wedge Y_i, sup)$ or $p = (A, sup)$ if $p$
        contains class label $Y_i$ then
            add tuple ($A$, $Y_i$, $sup$, $sup/patternset[A]$ ) into
            $RuleSet$
end
```

The Algorithm 1 has twice scans on the frequent patterns set. The first traversal is used for computing the classification rule confidence, while the second traversal finds all the classification rules with support and confidence parameters. 14 classification rules in Example 1 are obtained as Table II ($sup = 2/7$).

TABLE II
Classification rules mined in example 1

Rule ID	Rules	Support	Confidence
r_{01}	$a_1 \rightarrow Y_3$	0.29	0.41
r_{02}	$a_1, d_3 \rightarrow Y_3$	0.29	1.00
r_{03}	$a_1, b_2 \rightarrow Y_3$	0.29	0.67
r_{04}	$a_1, b_2, d_3 \rightarrow Y_3$	0.29	1.00
r_{05}	$d_3 \rightarrow Y_3$	0.29	0.67
r_{06}	$b_2, d_3 \rightarrow Y_3$	0.29	1.00
r_{07}	$b_2 \rightarrow Y_3$	0.29	0.67
r_{08}	$a_2 \rightarrow Y_1$	0.29	1.00
r_{09}	$a_1 \rightarrow Y_2$	0.29	0.41
r_{10}	$a_1, c_1 \rightarrow Y_2$	0.29	0.67
r_{11}	$c_1 \rightarrow Y_2$	0.29	0.67
r_{12}	$c_2 \rightarrow Y_1$	0.29	1.00
r_{13}	$a_2, c_2 \rightarrow Y_1$	0.29	1.00
r_{14}	$b_3 \rightarrow Y_1$	0.29	0.67

The classification rules generated by Algorithm 1 can be the raw rules set as the input of CRNN model construction. However, the number of the classification rules is usually very large. Therefore, it need to select the high quality rules for classification. There are many selection strategies in the previous works [5][12][13]. In this paper, the database coverage method which is proposed in [2] is used to select the classification rules.

Given two rules R_1 and R_2, R_1 is said having higher rank than R_2, if and only if (1) $conf(R_1) > conf(R_2)$; (2) if $conf(R_1) = conf(R_2)$, but $sup(R_1) > sup(R_2)$; (3) if $conf(R_1) = conf(R_2)$ and $sup(R_1) = sup(R_2)$, but R_1 has less items than R_2. According to this rules ranking methodology, the classification rules are sorted in descending order firstly. Every rule is tested for how many records are covered in the data set. The rules that cover at least one record are selected until all the data records are covered with a user predefined minimum threshold.

The new rule set with the rules selection strategy is much smaller than the original one. If the minimum data set coverage is set to 2 in Example 1, only 3 rules in Table II are remained, that are $r_{13} : a_2, c_2 \rightarrow Y_1$, $r_{04} : a_1, b-2, d_3 \rightarrow Y_3$ and $r09 : a_1 \rightarrow Y_2$.

B. FOIL Based Rule Generation

FOIL is used to find the rules that distinguish the positive examples from the negative ones [11]. Usually, FOIL is applied on each class when the data set has multiple class labels. The multiple class problems are transformed into several binary class problems. Rules are obtained for every class and then the rules for each class are merged to form the final rule set of the whole data set. The rules generated by the FOIL do not fit with the CRNN model since the rules lack the support and confidence parameters. The support and confidence need to be appended additionally.

In FOIL based rule generation procedure, a measurement is required to constrain how to select an attribute value to form a rule. The FOIL gain is usually used to get the information gain when an attribute value, A_i, is added to the current rule r. Let $|P|$ be the number of the positive examples and $|N|$ is the number of the negative examples. Once an attribute value, A_i, is added to the r's body, we get the new numbers of the positive and negative examples, as $|P'|$ and $|N'|$. Thus the FOIL gain of A_i is computed as following.

$$fgain(A_i) = |P'| \left(\log \frac{|P'|}{|P'| + |N'|} - \log \frac{|P|}{|P| + |N|} \right) \quad (3)$$

The attribute value with maximum FOIL Gain is selected to append the rule body until all the positive examples are covered. FOIL based rule generation is presented in Algorithm 2.

Algorithm 2: FOIL based rule generation. *Foil-rg*

```
Data: A data set, $ds$; the minimum foil gain threshold, $\delta$.
Result: A rule set, $RuleSet$.
begin
    initial $RuleSet = \emptyset$, given class labels set
    $Y = Y_1, \cdots, Y_M$ for each $c$ in $Y$ do
        initial $rs4c = \emptyset$ for class $c$
        get positive and negative data sets, $PD$, $ND$ of $c$
        while $PD \neq \emptyset$ do
            $rbody = \emptyset$, $PD' = PD$, $ND' = ND$
            while $|ND'| > 0$ and
            $rbody.length < max - rule - len$ do
                searching $A_i$ with maximum $fgain$
                if $fgain(A_i) < \delta$ then
                    break
                adding $A_i$ into the $rbody$
                delete examples not satisfied $rbody$ in $PD'$,
                $ND'$
            adding the $rbody$ into $rs4c$
            delete all the examples satisfied $rbody$ in $PD$
        for each $rb$ in $rs4c$ do
            computing the $sup(r)$ and $conf(r)$ of rule
            $r : rb \rightarrow c$
            adding rule tuple ($rb$, $c$, $sup(r)$, $conf(r)$) into
            $RuleSet$
    end
```

Four rules are generated from the data set shown in Example 1 where the *fgain* is set to 0.5. The rules set are $\{a_2 \rightarrow Y_1,\, b_3, c_3 \rightarrow Y_1,\, c_1 \rightarrow Y_2,\, d_3, a_1 \rightarrow Y_3\}$.

IV. CRNN Modelling

According to the ANN classification, we define the CRNN model structure. The CRNN structure relies on the rules set found in data set.

A. CRNN Model Design

The multilayer ANN structure is common seen in many studies. User usually needs to determine the input and output nodes depended on the problem and the number of nodes in the hidden layer as well. Thus a parameter learning algorithm is used to learn the weight parameters between the nodes in ANN. Similarly, the CRNN model is designed as a four layers neural network which includes the input layer (IN layer), pattern and class middle layers (PN and CN layer) and output layer (ON layer). We call the hidden layer as a middle layer for that the nodes in the middle layers of CRNN can be interpreted as actual meaning. The structure of CRNN is shown in Fig. 3.

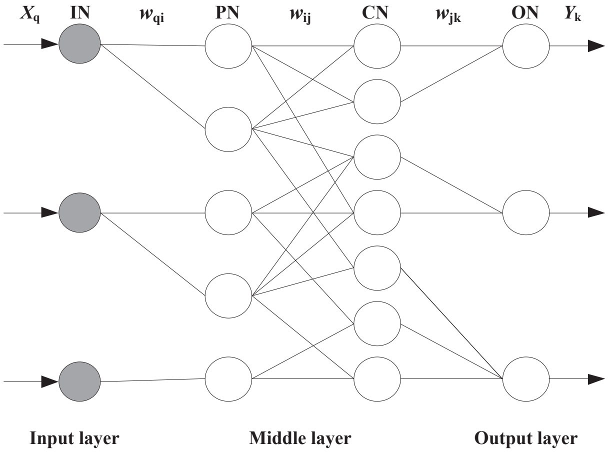


Fig. 3. Structure of CRNN model

In comparison with traditional ANN, There are some differences in the CRNN model.

- The nodes between the layers are partially connected while the nodes in ANN are not. In particular, PN node has and only has a input link, and CN node has and only has a output link.
- The nodes in the middle layers of CRNN stand actual meaning. Furthermore, the nodes and links between the middle layers are determined by classification rules set obtained from the training data set. Different dataset falls into different CRNN model structures automatically.
- The parameters, X_q, w_{qi}, w_{ij}, w_{jk} and Y_k, in CRNN require no complex and time-consuming learning algorithms (such as back propagation) to be computed out. These parameters are all obtained from the classification rules set.

B. CRNN Construction Using Rule Set

Every classification rule actually contains five data elements: the rule id, attribute items, class label, support and confidence, which is written as: $\{RID : (set\_of\_items, class, sup, conf)\}$. The mapping between the elements of a rule and the nodes of the four layers CRNN structure are shown as Fig. 4.

The detail mapping steps are described as following.

- The input nodes in CRNN stand for the attributes of the data record. That means the dimension of data set is the number of the input nodes. For example, the supermarket retail data can be presented by only one node in CRNN. The data set in Example 1 has four attributes, A, B, C, D, thus four input nodes appear in the CRNN.

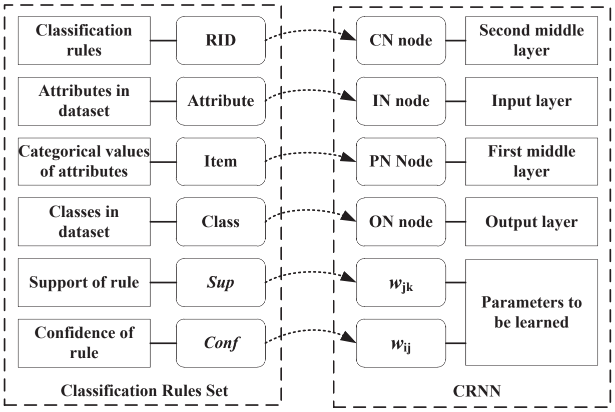


Fig. 4. Data mapping between rules set and CRNN

- The nodes in the first middle layer contain all the items that appear in the rule body. Every input node has links to its own attribute value.
- The nodes in the second middle layer correspond to the classification rules. Every rule is represented as one node in the CN layer. The connections between the PN layer and CN layer depend on whether the items in PN and CN layers are contained in a same rules.
- The output nodes represent the class labels of the data set, and every node in the CN layer only links to its class node in the ON layer.
- w_{jk} and w_{ij} are the support and confidence of the rule respectively. w_{ij} in CRNN gives the confidence of the rule in CN when the items in the rule body appear in PN node. w_{jk} represents the linkage of rule CN_j and class Y_k.
- w_{qi} is always 1.0 which means the attribute X_q has items of attribute value in PN_i.

Based on the above steps of CRNN structure building and parameter setting, the CRNN model construction is presented in Algorithm 3.

Given a classification rule set, $RuleSet = \{r_1, r_2, \cdots, r_R\}$, every rule r has n_r items in the rule body, the class label $Y_r \in \{Y_1, Y_2, \cdots, Y_M\}$. The rule r in RuleSet is formed as $(RuleID_r, P_r, Y_r, sup_r, conf_r)$. The CRNN is represented by a graph data structure $G = (V, E)$.

Example 2 (CRNN Construction) 14 rules in Example 1 are used to construct a CRNN model by Algorithm 3.

The first rule is $r_{01} : a_1 \rightarrow Y_3$. The nodes, X_A, PN_{a_1}, $CN_{r_{01}}$ and ON_{Y_3}, the linkages (X_A, PN_{a_1}, 1.0), (PN_{a_1}, $CN_{r_{01}}$, 0.41) and ($CN_{r_{01}}$, ON_{Y_3}, 0.29) are created. For the rule $r_{02} : a_1, d_3 \rightarrow Y_3$, the node X_A and ON_{Y_3} have been exists, thus only the nodes, X_D, PN_{d_3} and $CN_{r_{02}}$ need to be appended into the network. The edge between X_A and PN_{a_1} has been already in the CRNN. The links (X_D, PN_{d_3}, 1.0), (PN_{a_1}, $CN_{r_{02}}$, 1.00), (PN_{d_3}, $CN_{r_{02}}$, 1.0) and ($CN_{r_{02}}$, ON_{Y_3}, 0.29) are added. All the nodes and edges eventually are obtained in CRNN until all the rules have been processed. We will get to the final neural network structure as shown in Fig. 5.

Algorithm 3: CRNN model construction. *CRNNCON*

```
Data: A rule set generated from data set, $RuleSet$.
Result: The CRNN network, $G$.
begin
    initial $G =< V, E >, V = \emptyset, E = \emptyset$, where $V$ is the nodes set and $E$ is the links set with weights.
    for each $r$ in $RuleSet$ do
        $r = (RuleID_r, P_r, Y_r, sup_r, conf_r)$ and $p_r = i_1^r \wedge \cdots \wedge i_{m_r}^r$ if node of $Y_r$ not in $V$ then
            create node $ON_r$, and add it into $V$
        if node of $RuleID$ not in $V$ then
            create node $CN_r$, and add it into $V$
        create link($CN_r$, $ON_r$, $sup_r$) and add it into $E$
        for each $e$ in $P_r$ do
            if node of attribute that contains $e$ not in $V$ then
                create node $X_e$, and add it into $V$
            if node $e$ not in $V$ then
                create node $PN_e$, and add it into $V$
            if link ($X_e$, $PN_e$, 1) not in $E$ then
                create link ($X_e$, $PN_e$, 1.0) and add it into $E$
            create link ($PN_e$, $CN_r$, $conf_r$) and add it into $E$
    normalize all the weights in every output node
end
```

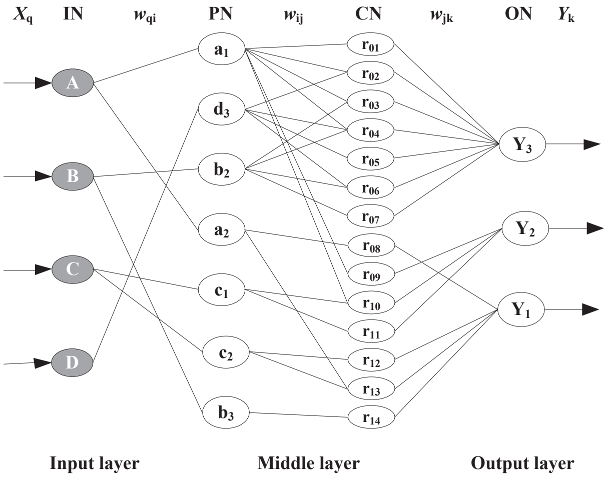


Fig. 5. The CRNN instance in example 2

The construction of the CRNN takes $|RQ|$ time in CRNNCON algorithm relying on the rules set size and rule body length. the R is the number of rule set and Q is the length of items in rule set, that $Q = n_1 + n_2 + \cdots + n_R$. Through the above description, we get the following characteristics of the CRNN classification model.

- *Structural autonomy*: CRNN connections between the nodes are determined by association rules, and IN nodes and PN nodes are fixed by the data itself naturally. Input layer of CRNN has the same number of the nodes as the dimension of the data set attributes. The PN nodes in the first middle layer are related to the items in the association rules. There are identical CN nodes to the rules. CRNN network can be fixed without user manual design contrasting with the traditional multi-layer neural network.

- *Parameter setting*: The parameters in the CRNN network are assigned directly. The weights between the IN nodes and PN nodes are Boolean type. If the item of the PN node, which belongs to a attribute, appears in an association rule, thus the weight between the input node and the PN node is set to 1, otherwise it is set to 0 (the links in current building process and examples are not considered). The weights, w_{ij} and w_{jk}, are also from the association rules. CRNN parameters are not obtained by learning algorithms as the traditional neural network, which makes the CRNN construction quickly.
- *Interpretable relationship*: The weights w_{ij} describe the relations between patterns and classes. This is identical to the meaning of the classification rule, that when the pattern appears, the class label will be predicted with confidence. The relations between the CN and ON nodes are obvious that the class labels are predicted correctly with the probabilities. The supports of the rules just represent the frequency in the data set (approximate with probability).
- *Transparent layers*: As we know, the hidden layer in the traditional neural network is design by user and the meaning of the hidden node is usually hard to be interpreted for user. But this situation is changed in the CRNN. The hidden layer which is called middle layer in the CRNN has intuitive explanation and is transparent to the user.

V. Prediction Using CRNN

This section introduces a CRNN classifier for the new data object prediction.

A. Building a Classifier

There are many activation functions used in the traditional ANN, such as threshold function, piecewise linear function, sigmoid function, Gaussian function and so on. A piecewise linear function is used in this paper, which is defined as $F: y = x, x \in (0,1]$, as shown in Fig. 6(a). The parameters of CRNN can not be negative, and the maximum value of x is 1.0.

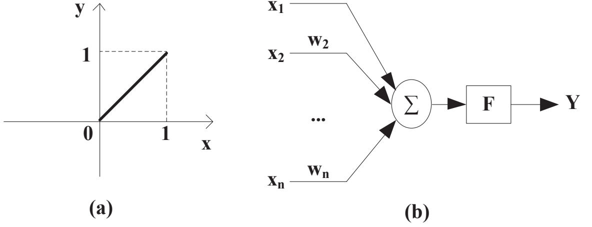


Fig. 6. CRNN activation function and node computing

General speaking, the node computing model in the neural network is shown as in Fig. 6(b). Given a neural node has n input links, $X = (x_1, x_2, \cdots, x_n)^T$, and the weights for every link are represented as $W = (w_1, w_2, \cdots, w_n)^T$. The output Y is computed as following.

$$Y = F(W^T X + b) = F\left(\sum_{i=1}^{n} w_i x_i + b\right) = \sum_{i=1}^{n} w_i x_i + b \tag{4}$$

where the b is the bias and F is the activation function. Let the rules set be $RuleSet = \{r_1, r_2, \cdots, r_R\}$, the class labels set $\{Y_1, Y_2, \cdots, Y_M\}$, and each class Y_k has n_k rules, that are $r_1^{Y_k}, r_2^{Y_k}, \cdots, r_{n_k}^{Y_k}$ $(1 \leq k \leq M)$, and $R = \sum_{k=1}^{M} n_k$. Every rule $r_p^{Y_k}$ has the support as $s_p^{Y_k}$, the confidence as $s_p^{Y_k}$. The items in the rule body are assumed as $\{I_1^p(Y_k), I_2^p(Y_k), \cdots, I_m^p(Y_k)\}$. Then the output node Y_k can be computed according to the above node computing model.

$$\begin{aligned} Y_k = \sum_{t=1}^{n_k} F(r_t) w_t + b_k^c \\ = s_1^{Y_k} F(r_1) + s_2^{Y_k} F(r_2) + \cdots + s_{n_k}^{Y_k} F(r_{n_k}) + b_k^c \end{aligned} \tag{5}$$

$F(r_t)$ is the output of the node in the CN layer, and $F(r_t)$ is obtained as following.

$$\begin{aligned} F(r_t) = conf(I_1^t(Y_k))F(I_1^t(Y_k)) + \cdots \\ +conf(I_{m_t}^t(Y_k))F(I_{m_t}^t(Y_k)) \\ = c_1^{Y_k}(r_t)S(I_1^t) + c_2^{Y_k}(r_t)\psi(I_2^t) + \cdots \\ +c_{m_t}^{Y_k}(r_t)\psi(I_{m_t}^t) + b_k^c \end{aligned} \tag{6}$$

Since the weights between the IN layer and PN layer are Boolean values, ψ is indicator function that ψ is 1 when PN node has output, otherwise ψ is 0. Assuming a Boolean vector, $V_t = (1, 1, \cdots, 1, 0 \cdots, 0)$, is the input vector of the CN layer where the element with zero value stands for there is no link to CN node. Thus the value of r_t is obtained as:

$$F(r_t) = (c_1^{Y_k}, c_2^{Y_k}, \cdots, c_{m_t}^{Y_k}, 0, \cdots, 0)V_t^T + b_t^r \tag{7}$$

We take this value into the Y_k, then we will get the Y_k.

$$\begin{aligned} Y_k = s_1^{Y_k}\{c_1^{Y_k}(r_1)\psi(I_1^1) + \cdots + c_1^{Y_k}(r_1)\psi(I_{m_1}^1) + b_1^r\} \\ +s_2^{Y_k}\{c_2^{Y_k}(r_2)\psi(I_1^2) + \cdots + c_2^{Y_k}(r_2)\psi(I_{m_2}^2) + b_2^r\} \\ + \cdots + \\ s_{n_k}^{Y_k}\{c_{n_k}^{Y_k}(r_{n_k})\psi(I_1^{n_k}) + \cdots + c_{n_k}^{Y_k}(r_{n_k})\psi(I_{m_{n_k}}^{n_k}) \\ +b_{n_k}^r\} + b_k^c \\ = (s_1^{Y_k} c_1^{Y_K}(r_1), \cdots, s_1^{Y_k} c_1^{Y_K}(r_1), 0, \cdots, 0)V_1^T + s_1^{Y_k} b_1^r \\ +(s_2^{Y_k} c_2^{Y_K}(r_2), \cdots, s_2^{Y_k} c_2^{Y_K}(r_2), 0, \cdots, 0)V_2^T + s_1^{Y_k} b_2^r \\ + \cdots + \\ (s_{n_k}^{Y_k} c_{n_k}^{Y_K}(r_{n_k}), \cdots, s_{n_k}^{Y_k} c_{n_k}^{Y_K}(r_{n_k}), 0, \cdots, 0)V_{n_k}^T \\ +s_{n_k}^{Y_k} b_{n_k}^r + b_k^c \end{aligned} \tag{8}$$

$V_1, V_2, \cdots, V_{n_k}$ is normalized by unify order. E.g. one sorts the rules by the rules' number. A new Boolean vector U from V is got according to the rules order. The final output value of class, Y_k, is represented as following.

$$Y_k = \left(s_1^{Y_k}c_1^{Y_k}, s_2^{Y_k}c_2^{Y_k}, \cdots, s_{n_k}^{Y_k}c_{n_k}^{Y_k}\right)\begin{pmatrix} U_1 I \\ U_2 I \\ \cdots \\ U_{n_k} I \end{pmatrix} + \left(s_1^{Y_k}, s_2^{Y_k}, \cdots, s_{n_k}^{Y_k}\right)\begin{pmatrix} b_1^r \\ b_2^r \\ \cdots \\ b_{n_k}^r \end{pmatrix} + b_k^c \tag{9}$$

where I is a unit column vector, namely $I^T = (1, 1, \cdots, 1)$. According to the structure of the CRNN, $U_t I$ is just the number of the items in the rule r_t. When the new data comes, the above method is used to compute Y value for every class. The class label with the maximum value of Y is selected as the class label of the new data. However, one problem is still in the formula (9) because the parameters of b are unknown. In this paper, we set all the parameter of b to zero. The final computing model of Y_k is shown as following formula.

$$Y_k = \left(s_1^{Y_k}c_1^{Y_k}, s_2^{Y_k}c_2^{Y_k}, \cdots, s_{n_k}^{Y_k}c_{n_k}^{Y_k}\right)\begin{pmatrix} U_1 I \\ U_2 I \\ \cdots \\ U_{n_k} I \end{pmatrix} \tag{10}$$

All the s and c are known for us and the U can be known if user gives the new data. The class k with maximum of Y_k is regarded as the class label of the new data, which is shown as formula 11.

$$k_{predict} = \underset{k}{\operatorname{argmax}}\{Y_k\}, k \in \{1, 2, \cdots, M\} \tag{11}$$

Example 3 (CRNN Classification) Let the new data be $x = (A = a_1, B = b_2, C = c_2, D = d_1)$, predicting the class label of x using the CRNN model as shown in Fig. 5. All the values of Y are calculated as following.

$$Y_1 = \left(s_8^{Y_1}c_8^{Y_1}, s_{12}^{Y_1}c_{12}^{Y_1}, s_{13}^{Y_1}c_{13}^{Y_1}, s_{14}^{Y_1}c_{14}^{Y_1}\right)\begin{pmatrix} U_8 I \\ U_{12} I \\ U_{13} I \\ U_{14} I \end{pmatrix} = 0.435 \tag{12}$$

The computing processes of Y_2 and Y_3 are same as Y_1. The Y_2 and Y_3 are 0.264 and 1.085 relatively. The maximum value is Y_3, so the class of the new data x is Y_3.

B. General Form of Rule-based Decision Methodology

It is known that the w_{ij} is $conf(r)$ and w_{jk} is $sup(r)$ in Algorithm 3. The value of Y_k is transformed as following when the w_{ij} is set to be $conf(r)/L$, where L is the length of the rule body.

$$\begin{aligned} Y_k &= \left(s_1^{Y_k}\frac{c_1^{Y_k}}{L}, s_2^{Y_k}\frac{c_2^{Y_k}}{L}, \cdots, s_{n_k}^{Y_k}\frac{c_{n_k}^{Y_k}}{L}\right)\begin{pmatrix} U_1 I \\ U_2 I \\ \cdots \\ U_{n_k} I \end{pmatrix} \\ &= \left(s_1^{Y_k}c_1^{Y_k}, s_2^{Y_k}c_2^{Y_k}, \cdots, s_{n_k}^{Y_k}c_{n_k}^{Y_k}\right) \end{aligned} \tag{13}$$

where $U_t I$ is just the number of the items in the rule r_t. This is a kind of general form for the rule based decision process. Let us exam the three methods in Fig. 1, and we can get that they are the special forms of the above decision expression where the node activation function F is set as threshold type. We assume the CN node is in activation when all the PN nodes (items of rule body) have output, which means the rule is fully satisfied as in [4][5][14].

- We treat the Y_k (sum of $s_i^{Y_k}c_i^{Y_k}$) in another way, not in sum expression. All $s_i^{Y_k}c_i^{Y_k}$ within Y_k are sorted in descending. The class label with first highest value $s_i^{Y_k}c_i^{Y_k}$ is selected which is the first method described as in [4]
- For the second method in Fig. 1, the satisfied *top-K* rules are selected for each class Y_k, and the class label with highest Y_k value of the K rules (i.e. $K = 5$) is treated as predicting class. For example, the Laplace accuracy is similar with the confidence when the data set size is big enough. The class label with highest arithmetic average of Laplace accuracy of the *top-K* rules is regarded as the predicting tag in [14]. The Y_k here is the weighted average of confidence (the supports in Y_k is the weights sum of confidences, because the sum of the supports for Y_k is 1.0 in Algorithm 3 [line 15]).
- Let the rule be $r : p \rightarrow c$, the sum of $s_i^{Y_k}c_i^{Y_k}$ ($sup(r)conf(r)$ for short) is equal as following: $Y_k = \sum sup(r)conf(r) = \sum sup(p, c)\frac{sup(p,c)}{sup(p)} = \sum \frac{sup^2(p,c)}{sup(p)}$ The above Y_k has similar function with the sum of χ^2 for each rule in the group [5].

VI. Experimental Results

In this paper, 21 data sets from UCI ML Repository are used to evaluate our approach. Discretization of continuous attributes is done using the same method in [15]. We have conducted the accuracy study on these data and compared CRNN with C4.5 [6] and CPAR [14]. The CRNN algorithm is implemented in Python v2.7, and all the other approaches are tested by their authors. All experiments are conducted on a desktop computer with Intel Core 2 CPU of 2.80GHz, 4GB memory and Windows 7. The rules sets are generated by two different ways, which are ARM based rule generation (CRNN-a for short) and FOIL based rule generation (CRNN-f for short). The performance of CRNN on the two rule sets is shown as in Table III.

The parameters of the CRNN model are set as following. The support and confidence of association rule mining are set to 0.05 and 0.9. The database coverage threshold is set to 10. With regard to the FOIL based rule generation, the fgain

TABLE III

ACCURACY: C4.5, CPAR, CRNN-A AND CRNN-F

ID	Dataset	C4.5	CPAR	CRNN-c	CRNN-f
01	austra	84.7	86.2	**86.5**	85.9
02	breast	95.0	96.0	**97.3**	95.4
03	cleve	78.2	81.5	84.2	**84.5**
04	crx	84.9	85.7	**86.5**	85.9
05	diabetes	74.2	75.1	**75.4**	73.7
06	german	72.3	**73.4**	71.0	72.0
07	glass	68.7	**74.4**	65.5	66.9
08	heart	80.8	82.6	84.3	**84.8**
09	hepatic	80.6	79.4	**84.4**	81.3
10	horse	82.6	84.2	79.6	**84.8**
11	hypo	**99.2**	98.1	95.2	99.1
12	iris	95.3	94.7	94.7	**96.0**
13	labor	79.3	84.7	**90.0**	77.7
14	led7	73.5	73.6	**73.8**	62.0
15	lymph	73.5	82.3	82.4	**83.1**
16	pima	**75.5**	73.8	73.2	72.5
17	sick	**98.5**	96.8	93.9	96.3
18	sonar	70.2	79.3	80.3	**81.2**
19	wave	78.1	80.9	78.9	**83.0**
20	wine	92.7	**95.5**	87.6	92.2
21	zoo	92.2	**95.1**	90.1	90.1
Avg.		82.38	84.44	83.56	83.26
#top-1		3	4	7	7

is set to 0.1. As the result shown in Table 4, the average accuracy of CRNN is better than C4.5. CRNN-a and CRNN-f have higher performance in some data set which are shown as bold font in Table III.

TABLE IV

EFFICIENCY OF CPAR, CRNN-A AND CRNN-F

Average	CPAR		CRNN-c		CRNN-f	
	Time	Rules	Time	Rules	Time	Rules
Arithmetic	0.2	261.4	85.0	55.1	14.4	26.3
Geometric	0.04	105.6	19.4	39.8	1.7	19.7

Table IV shows the algorithm running efficiency (in sec.) and average number of rules used in CRNN. We compute the arithmetic and geometric average of the running time and rules for the CPAR, CRNN-a and CRNN-f on the 21 datasets.

VII. CONCLUSIONS

The rule base classification and neural network classification are two popular methods in machine learning and data mining. Association classification invokes a new aspect of classification methodology. This paper combines the classification rule and neural network as a classifier which takes advantages of rule base classification and neural network approaches. CRNN as the evaluation result achieves high accuracy and efficiency, which can be credited to the features of CRNN. Firstly, CRNN use network structure to make prediction, which takes all the rules and coupling relations between the rules into consideration other than a part of rules in the rule set. Secondly, the parameters in CRNN are all from the rules without the complex learning procedure. Therefore, the efficiency of this neural network can be obtained obviously. Moreover, CRNN has two good characteristics. CRNN holds a general expression of the rule based decision methodology. The nodes in CRNN are all transparent for users that we can get clearly interpretation of hidden nodes in CRNN which is hard for traditional neural network.

CRNN introduces a new approach towards efficient and high quality classification model integrating the classification rules and neural network. Our further work will focus on the following two tasks. The bias of node computing model is set to zero in this paper, and we will do more study on how to learn the bias parameters by a linear computing method. From the evaluation result, we know that the rule set is critical for CRNN. Better rule set makes higher accuracy of CRNN. We will do more research work on the effective rule set mining.

REFERENCES

[1] B. Bringmann, S. Nijssen and A. Zimmermann , “Pattern-based classification: a unifying perspective”, *In Proceedings of ECML-PKDD workshop on from local patterns to global models*, pp. 36-50, 2009

[2] Y. Zhao, C. Zhang and L.Cao, “Post-Mining of Association Rules: Techniques for Effective Knowledge Extraction”, *Information Science Reference-Imprint: IGI Publishing*, 2009.

[3] B. Liu, Y. Ma and C. Wong, “Improving an association rule based classifier”, *Principles of Data Mining and Knowledge Discovery*, pp. 293-317, 2000.

[4] B. Liu, W. Hsu and Y. Ma, “Integrating classification and association rule mining”, *In KDD98*, New York, NY, Aug. pp. 80-86, Aug. 1998.

[5] W. Li, J. Han and J. Pei, “CMAR: Accurate and efficient classification based on multiple class-association rules”, *In Data Mining ’01*, Proceedings IEEE International Conference on, pp. 369-376, Nov. 2001.

[6] J. Quinlan, “C4.5: programs for machine learning”, *Morgan kaufmann*, vol. 1, 1993.

[7] R. Agrawal and R. Srikant, “Fast algorithms for mining association rules”, *In Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, vol. 1215, pp.487-499, 1994.

[8] J. Han, J. Pei and Y. Yin, “Mining frequent patterns without candidate generation”, *In ACM SIGMOD Record*, vol. 29, no. 2, pp. 1-12, 2000.

[9] B.S. Prachitee and S.D. Sheetal, “A classification technique using associative classification learning”, *International Journal of Computer Applications*, vol. 20, no.5, pp. 20-28, 2011.

[10] D.E. Rumelhart, G.E. Hintont and R. J.,Williams, “Learning representations by back-propagating errors”, *Nature*, vol. 323, no. 6088, pp. 533-536, 1986.

[11] J. Quinlan and R. Cameron-Jones, “FOIL: A midterm report”, *In Machine Learning: ECML-93*, Springer Berlin/Heidelberg, pp. 1-20, 1993.

[12] W. Cohen, “Fast Effective Rule Induction”, *In Proceedings of the Twelfth Conference on Machine Learning*, pp. 115-123, Jul. 1995.

[13] A. Zimmermann and B. Bringmann, “CTC-correlating tree patterns for classification”, *In ICDM*, pp. 833-836, 2005.

[14] X. Yin and J. Han, “CPAR: Classification based on predictive association rules”, *In Proceedings of the Third SIAM International Conference on Data Mining*, vol. 3, pp. 331-335, 2003.

[15] U. Fayyad and K. Irani, “Multi-interval discretization of continuous-valued attributes for classification learning”, *IJCAI ’93*, pp, 1022-1027, 1993.