Building an Iris Plant Data Classifier Using Neural Network Associative Classification

Ms.Prachitee Shekhawat[1], Prof. Sheetal S. Dhande[2]

[1,2]Sipna's College of Engineering and Technology, Amravati, Maharashtra, India.

[1]prachitee.shekhawat@rediffmail.com

Abstract

Classification rule mining is used to discover a small set of rules in the database to form an accurate classifier. Association rules mining are used to reveal all the interesting relationship in a potentially large database. For association rule mining, the target of the discovery is not predetermined, while for classification rule mining there is one and only one predetermined target. These two techniques can be integrated to form a framework called Associative Classification method. The integration is done by focusing on mining a special subset of association rules called Class Association rules (CAR).This project paper proposes a Neural Network Association Classification system, which is one of the approaches for building accurate and efficient classifiers. Experimental result shows that the classifier build for iris plant dataset in this way is more accurate than previous Classification Based Association.

Keywords: *Data Mining, Association Rule Mining, Classification, Associative Classification, Backpropagation neural network.*

1. Introduction

Data mining is the analysis of large observational data sets to find unsuspected relationships and to summarise the data in novel ways that are both understandable and useful to the data owner. Data mining tools can forecast the future trends and activities to support the decision of people. It is a discipline, lying at the intersection of statistics, machine learning, data management and databases, pattern recognition, artificial intelligence, and other areas. It is often set in the broader context of knowledge discovery in databases (KDD). The KDD process consists of following stages: selecting the target data, pre-processing the data, transforming them if necessary, performing data mining to extract patterns and relationships, and interpreting and assessing the discovered structures.

Classification and association rule mining are two basic tasks of Data Mining. Association rules mining are capable of revealing all interesting relationships in a potentially large database. These relationships show strong associations between attribute-value pairs (or items) that occur frequently in a given data set. Set of association rules can be used not only for describing the relationships in the databases, but also for discriminating between different kinds or classes of database instances. Classification rules mining is kind of supervised learning algorithm which is capable of mapping instances into distinct classes. In other words, classification is a task of assigning objects to one of several predefined categories. It consists of predicting the value of a (categorical) attribute (the class) based on the values of other attributes (the predicting attributes).

Association rule mining and classification rule mining can be integrated to form a framework called as Associative Classification and these rules are referred as Class Association Rules. The discriminating power of association rules can be used to solve classification problem. Here, the frequent patterns and their corresponding association or correlation rules characterize interesting relationships between attribute conditions and class labels. The general idea is that we can search for strong association between frequent patterns and class labels. By using the discriminative power of the Class Association Rules we can also build a classifier.

The association rule mining is only possible for categorical attributes. Hence, class association rules are restricted to problems where the instances can only belong to a discrete number of classes.

Data mining in the proposed, Neural Network Associative Classification, system thus consists of three steps:

1) If any, discretizing the continuous attributes,
2) Generating all the Class Association Rules (CARs), and
3) Building a classifier with the help of Backpropagation Neural Network based on the generated CARs set.

Here, we have to analyze the iris plant dataset and mine all the accurate association rule that will be use to build an efficient classifier on the basis of the following measurements: sepal length, sepal width, petal length, and petal width. The motivations of choosing this dataset are:

1) Botanical field is a general domain which has a great deal of effort in terms of knowledge management.
2) Contain addition, valuable information which is up-to-date and comprehensive.
3) Our system can more easily be adapted to this domain.

This system proposes a new way to build accurate classifier by using association rule mining techniques. The analysis of performance of the iris plant classifier is based on criteria: misidentified plants by testing set (accuracy). Experimental result show that classifier built this way are, in general, more accurate than the previous classification system

The paper is organized as follows: section 2 contains the brief introduction to major previous work done about data mining. Section 3 provide the description about applying the Backpropagation Neural Network to Associative Classification and section 4 presents our experimental setup for build an Iris plant classifier and discussed the result. Section 5 concludes the paper.

2. Literature Survey

Data Mining is categorised into different tasks depending on what the data mining algorithm under consideration is used for. A rough characterisation of the different data mining tasks can be achieved by dividing them into descriptive and predictive tasks [1]. A predictive approach tries to assign a value to a future or unknown value of other variables or database fields [1], whereas description tries to summarise the information in the database and to extract patterns. Association rule mining is a descriptive task. A rule out of the set of association rules is one descriptive pattern—a compact description for a very small subset of the whole data. Typical predictive tasks are classification and regression. Classification involves learning a function which is capable of mapping instances into distinct classes.

Regression maps instances to a real-valued variable. Therefore, a class association rule is obviously a predictive task.

One of the first problems with the term “data mining” is that it means different things to different audiences; lay use of the term is often much broader than its technical definition. A good description of what data mining does is: “discover useful, previously unknown knowledge by analyzing large and complex data sets”. Data mining itself is a relatively narrow process of using algorithms to discover predictive patterns in data sets. The process of applying or using those patterns to analyze data and make predictions is not data mining. A more accurate term for those analytical applications is “automated data analysis,” which can include analysis based on pattern queries (which involve identification of some predictive model or pattern of behaviour and searching for that pattern in data sets) or subject-based queries (which start with a specific and known subject and search for more information). Mary DeRosa [2] describes data mining and automated data analysis are the technique that has significant potential for use in countering terrorism but there is a principal reasons for public concern about these tools is that there appears to be no consistent policy guiding decisions about when and how to use them.

On all the important topics in data mining, such as classification, clustering, statistical learning, association analysis, and link mining, algorithms are presented in [3]. Presented algorithms are C4.5, *k*-Means, Support vector machines (SVM), Apriori, Expectation–Maximization(EM), PageRank, AdaBoost, *k*-nearest neighbor (*k*NN), Naive Bayes, and Classification and Regression Trees(CART). With each algorithm, there is a description of the algorithm, discussion on the impact of the algorithm, and review of the current and further research on the algorithm. But none of the data mining algorithms can be applied to all kinds of data types.

The current emphasis in Data Mining and Machine Learning research involves improving the classifiers to be more precise and efficient. The important classification algorithms are decision tree, Naive-Bayes classifier and statistics [3]. They use heuristic search and greedy search techniques to find the subsets of rules to find the classifiers. C4.5 and CART are the most well-known decision tree algorithms. Given a set S of cases, C4.5 first grows an initial tree using the divide-and-conquer algorithm [3].The CART decision tree is a binary recursive partitioning procedure capable of processing continuous and nominal attributes both as targets and predictors.

Another class of data mining algorithms is frequent pattern extraction. The goal here is to extract from the tabular data model all combinations of variable instantiations that exist in the data with some predefined level of regularity. Typically the basic kind of pattern to be extracted is an association: tuple of two sets, with a unidirectional causal implication between the two sets, A -> B. For a large databases, [4] describes an apriori algorithm that generate all significant association rules between items in the database. The algorithm makes the multiple passes over the database. The frontier set for a pass consists of those itemsets that are extended during the pass. In each pass, the support for candidate itemsets, which are derived from the tuples in the databases and the itemsets contain in frontier set, are measured. Initially the frontier set consists of only one element, which is an empty set. At the end of a pass, the support for a candidate itemset is compared with the minsupport. At the same time it is determined if an itemset should be added to the frontier set for the next pass. The algorithm terminates when the frontier set is empty. After finding all the itemsets that satisfy minsupport threshold, association rules are generated from that itemsets.

Bing Liu and et al. [5] had proposed a Classification Based on Associations (CBA) algorithm that discovers Class Association Rules (CARs). It consists of two parts, a rule generator, which is called CBA-RG, is based on Apriori algorithm [4] for finding the association rules and a classifier builder, which is called CBA-CB. In Apriori Algorithm, itemset (set of items) were used while in CBA-RG, ruleitem, which consists of a condset (a set of items) and a class. Class Association Rules that are used to create a classifier in is more accurate than C4.5 [6] algorithm. But the Classification Based on Associations (CBA) algorithm needs the ranking rule before it can create a classifier. Ranking depends on the support and confidence of each rule. It makes the accuracy of CBA less precise than Classification based on Predictive Association Rules [12].

Artificial neural network composed of a number of interconnected units. Each unit has input/output characteristics and implements a local computation or function. The output of any unit is determined by its input/output characteristics, its interconnection to other units, and possible external input. Although hand crafting of the network is possible, the network usually develops an overall functionality through one or more forms of training [7]. There are two types of neural network topologies: Feed forward networks and recurrent networks. This project uses various back propagation neural networks. It uses a feed forward mechanism, and is constructed from simple computational units referred to as neurons. Neurons are connected by weighted links that allow for communication of values. When a neuron’s signal is transmitted, it is transmitted along all of the links that diverge from it. These signals terminate at the incoming connections with the other neurons in the network. The backpropagation algorithm, also known as generalized delta rule. For one-hidden-layer backpropagation network, 2N+1 hidden neurons are required [8], where N is the number of inputs. In [9], Neural Network Associative Classification system is applied on lenses, iris plant, diabetes, and glass datasets. The classifier for iris plant dataset did not outperform as compare to CBA.

3. Applying a Backpropagation neural network to Associative Classifications

The proposed system, Neural Network Associative Classification, undergoes three steps: Pre-processing, generating Class Association Rules (CARs), and creating backpropagation neural network from Associative Classification. Neural Network Associative Classification architecture is shown in fig. 1.

3.1 Pre-processing

The main goal of this step is to prepare the data for the next step: Class Association Rule Mining. This step begins with the transformation process of the original dataset. After that, continuous attributes undergo discretization process.

3.1.1 Transformation

The data mining based on the neural network can only handle the numeric data so it is needed to transform the character date into numeric data.

3.1.2 Discretization

Classification datasets often contain many continuous attributes. Mining of association rules with continuous attributes is still a research issue. So our system involves discretizing the continuous attributes based on the pre-determined class target. For discretization, we have used the algorithm from [10].

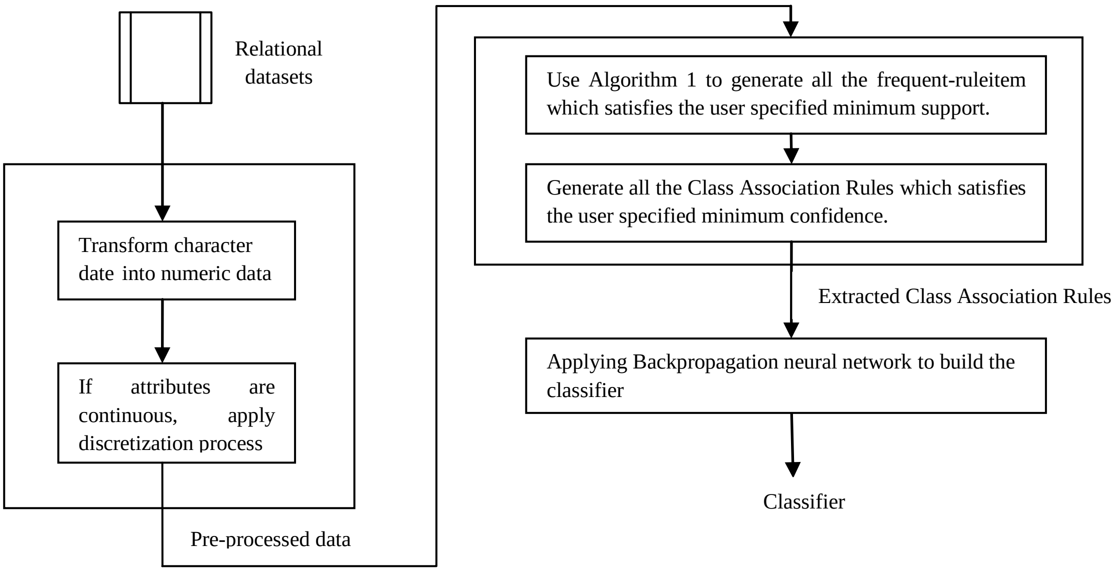

Relational datasets
Transform character date into numeric data
If attributes are continuous, apply discretization process
Pre-processed data
Use Algorithm 1 to generate all the frequent-ruleitem which satisfies the user specified minimum support.
Generate all the Class Association Rules which satisfies the user specified minimum confidence.
Extracted Class Association Rules
Applying Backpropagation neural network to build the classifier
Classifier

Fig 1: Neural network Associative Classification System Architecture

3.2 Generating Class Association Rules

This step presents a way for generating the Class Association Rules (CARs) from the pre-processed dataset.

3.2.1 Association Rules

Consider:

D= {d_1, d_2... d_n} is a database that consist of set of n data and d ⊆ I
I= {i_1, i_2… i_m} is a set of all items that appear in D

The association rule has a format, A→ B, by support=s%, confidence=c% and A ∩ B = ø and where A, B ⊆ I.

- Support value is frequency of number of data that consist of A and B or P(A ∪ B) and is given by

$$support, s(A \rightarrow B) = \frac{\sigma(A \cup B)}{N} * 100\% \qquad ...(1)$$

- Confidence is frequency of number of data that consist of A and B or P(B|A) and given by

$$confidence, c(A \rightarrow B) = \frac{\sigma(A \cup B)}{\sigma(A)} * 100\% \qquad ...(2)$$

3.2.2 Class Association Rules

The Class Association Rules are the subset of the association rules whose right-hand-side is restricted to the pre-determined target. According to this, a class association rule is of the form

$$A \rightarrow b_i$$

where b_i is the class attribute and A is $\{b_1, . . . , b_{i-1}, b_{i+1}, . . . , b_n\}$.

Consider:

D = $\{d_1, d_2... d_i\}$ is a database that consist of set of data that have n attribute and class label by d = $\{b_1, b_2... b_n, y_k\}$ where k= 1, 2...m.
I= $\{A_1, A_2\ldots A_m\}$ is a set of all items that appear in D
Y= $\{b_1, b_2\ldots b_m\}$ is a set of class label m class

A class association rule (CAR) is an implication of the form condset → b, where condset $\subseteq$ I, and b $\subseteq$ Y. A rule condset → b holds in D with support = s% and confidence = c% following these conditions:

- Support value is the frequency of number of all data that consist of itemset condset and class label b.
- Confidence value is the frequency of number of data that consist of class label b when data have itemset condset.

Before formulating the formulas for support and confidence, few notations are defined as:

- condsupCount is the number of data that consist of condset.
- rulesupCount is the number of data that consist of condset and has the label b.

Then

$$support = \frac{rulesupCount}{|D|} * 100\% \qquad ...(3)$$

$$confidence = \frac{rulesupCount}{condsupCount} * 100\% \qquad ...(4)$$

The algorithm 1 is shown below which is used to find the Class Association Rules:

```
$F_1$= {large 1-ruleitems};
$CAR_1$ = genRules($F_1$);
for (k=2; $F_{k-1}$=Ø; k++)
    $C_k$= candidateGen($F_{k-1}$);
    for each data case d ∈ D
        $C_d$ = ruleSubset($C_k$, d);
         for each candidate c ∈ $C_d$
            c.condsupCount++;
            if d.class=c.class then
                c.rulesupCount++;
           end
       end
    end
    $F_k = \{ c \in C_k \mid c.\text{rulesupCount} \geq \text{minsup}\}$;
    $CAR_k = \text{genRules}(F_k)$;
end
$CARs = \cup_k CAR_k$;
```

Algorithm 1: Proposed Algorithm

An example of searching the Class Association Rules following the Algorithm 1 is given below in Table 1.

Table 1: Example data.

	A	B	C
1.	a1	b1	c1
2.	a1	b2	c1
3.	a2	b1	c1
4.	a3	b2	c2
5.	a2	b2	c2
6.	a1	b1	c2
7.	a1	b1	c1
8.	a3	b3	c1
9.	a3	b3	c2
10.	a2	b3	c2

- A and B are predicted attributes. A have the possible values of a1, a2, and a3 and B's possible values are b1, b2, and b3.
- C is class label and possible values are c1 and c2.
- Minimum support(minsup) = 15%
- Minimum confidence(minconf) = 60%

In cases where a ruleitem is associated with multiple classes, only the class with the largest frequency is considered by current associative classification methods. We can express details as in Table 2. For CAR, condset →y, the format for ruleitem is

<(condset, condsupCount),(y, rulesupCount)>

Consider for example the ruleitem <{(A,a1)},(C,c1)>, it means that the frequency of condset (A,a1) in data is 4,it is also called as condsupCount, and the frequency of (A,a1) and (C,c1) occurring together is 3, also known as rulesupCount. So, the support and confidence are 30% and 75% respectively by using equations 3 and 4. As it satisfy the minsup threshold ruleitem is a frequent 1-ruleitem (F_1). As the confidence of this ruleitem is greater than minconf threshold, Class Association Rule (CAR_1) is formed as

$$(A, a1) \rightarrow (C, c1)$$

Repeat this procedure and generate all F_1 ruleitems. Using F_1 ruleitem form Candidate 2-ruleitms C_2 in the same way. Class Association Rules that are generated from table 1 is shown in table 2 and it can be express as:

1) (A,a1) → (C, c1) s=30%, c=75%
2) (A,a2) → (C, c2) s=20%, c=66.67%
3) (A,a3) → (C, c2) s=20%, c=66.67%
4) (B,b1) → (C, c1) s=30%, c=75%
5) (B,b2) → (C, c2) s=20%, c=66.67%

6) (B,b3) → (C, c2) s=20%, c=66.67%

7) {(A,a1),(B,b1)} → (C, c1) s=20%, c=66.67%

where s and c are support and confidence value respectively.

Table 2: Generating CARs following proposed algorithm

1st pass	F_1	<({(A,a1)},4),((C,c1),3)>
		<({(A,a2)},3),((C,c2),2)>
		<({(A,a3)},3),((C,c2),2)>
		<({(B,b1)},4),((C,c1),3)>
		<({(B,b2)},3),((C,c2),2)>
		<({(B,b3)},3),((C,c2),2)>
CAR_1	(A,a1)→(C,c1)	
	(A,a2)→(C,c2)	
	(A,a3)→(C,c2)	
	(B,b1)→(C,c1)	
	(B,b2)→(C,c2)	
	(B,b3)→(C,c2)	
2nd pass	**C_2**	<{(A,a1),(B,b1)},(C,c1)>
		<{(A,a1),(B,b2)},(C,c1)>
		<{(A,a2),(B,b1)},(C,c1)>
		<{(A,a2),(B,b2)},(C,c2)>
		<{(A,a2),(B,b3)},(C,c2)>
		<{(A,a3),(B,b2)},(C,c2)>
		<{(A,a3),(B,b3)},(C,c1)>
		<{(A,a3),(B,b3)},(C,c2)>
	F_2	<({(A,a1),(B,b1)},3),((C,c1),2)>
CAR_2	{(A,a1),(B,b1)} → (C,c1)	
CARs	$CAR_1 \cup CAR_2$	

3.1 Creating Backpropagation Neural Network from Associative Classification

Backpropagation Neural Network is a feedforward neural network which is composed of a hierarchy of processing units, organized in a series of two or more mutually exclusive sets of neurons or layers. The first layer is an input layer which serves as the holding site for the inputs applied to the neural network. The basic role of this layer is to hold the input values and to distribute these values to the units in the next layer. The last, output, layer is the point at which overall mapping of the network input is available. Between these two layers lies zero or more layer of hidden units. In this internal layer, addition remapping or computing takes place.

Generally Backpropagation Neural Network use logistic sigmoid activation function when output is required in the range [0, 1]. The output at each node can be defined in this equation.

$$outputj = \sigma(netj) = \frac{1}{(1-e^{-netj})} \qquad \ldots (5)$$

By

$$netj = \sum_{i=0}^{n} w_{ji}\, x_i \qquad \ldots (6)$$

Where, w_{ji} is the connection strength from the node i to node j and x_i is the output at node i.

The structure of Backpropagation neural network that is created with input as Class Association Rules is shown in fig. 2.

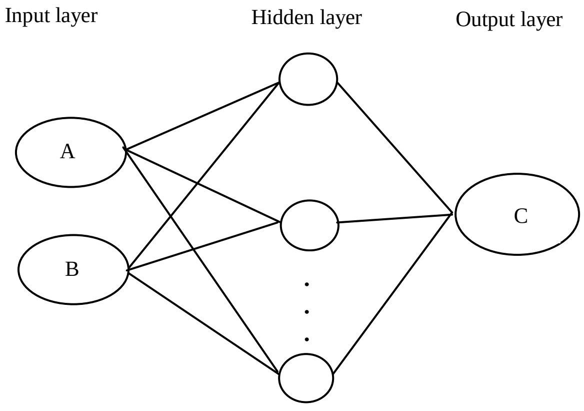


Fig. 2: Structure of Backpropagation Neural Network from Class Association Rules

Suppose that ruleitem for learning is {(A, a1), (B, b1)} → (C, c1). These character data need to be transformed into numeric data and so A with values a1, a2, and a3 are encoded with numeric value as 1, 2, and 3 respectively. Similarly B with values b1, b2, and b3 are encoded with numeric value as 1, 2, and 3 respectively. The class label c1 is denoted by 1 and c2 by 0 and now we will have the input is “1 1” and output will be “1” for the above mention ruleitem. This is shown in fig. 3.

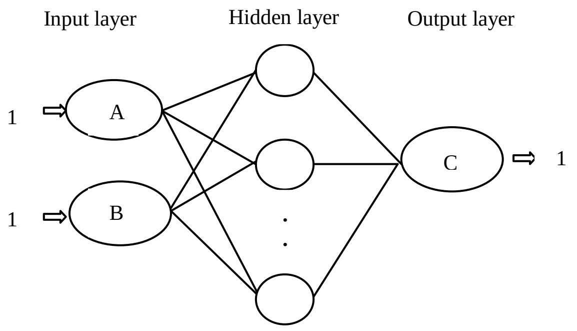


Fig. 3: Structure of Backpropagation Neural Network for “{(A, a1), (B, b1)} → (C, c1)” as an input.

4. Experimental setup and results

The Neural Network Associative Classification system is a user-friendly application developed in order to classify the relational dataset. This system is essentially a process consisting of three operations:

1) Loading the relational (table) dataset,

2) Discretizing continuous attributes, if any
3) Let the user enters the two-threshold values support and confidence,
4) Let the system perform the operations of calculating CAR$_s$ (Class Association Rules),
5) Let the system form a neural network, with the inputs as CARs set, by using Backpropagation algorithm,
6) Let the system perform the testing on the network, and
7) Let the user enter the unknown input and the system will provide the predicted class based on the training.

The discriminating feature of this system is that it uses the CARs to train the network to perform the classification task. So the system will render the efficient and accurate class based on the predicted attributes.

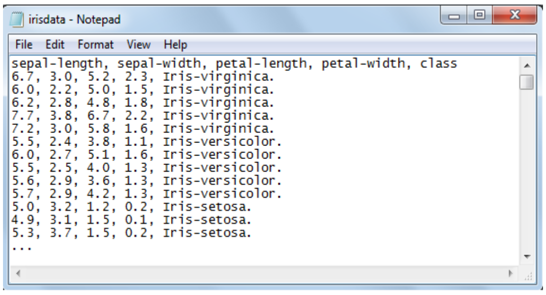
4.1 Data Description

This project makes use of the well known Iris plant dataset from UCI repository of Machine Learning Database [11], which refers to 3 classes of 50 instances each, where each class refers to a type of Iris plant. The first of the classes is linearly distinguishable from the remaining two, with the second two not being linearly separable from each other. The 150 instances, which are equally separated between the 3 classes, contain the following four attributes: sepal length and width, petal length and width. A sepal is a division in the calyx, which is the protective layer of the flower in bud, and a petal is the divisions of the flower in bloom. The minimum values for the raw data contained in the data set are as follows (measurements in centimetres): sepal length (4.3), sepal width (2.0), petal length (1.0), and petal width (0.1). The maximum values for the raw data contained in the data set are as follows (measurements in centimetres): sepal length (7.9), sepal width (4.4), petal length (6.9), and petal width (2.5). Each instance will belong to one of the following class: Iris Setosa, Iris Versicolour, or Iris Virginica.

The proposed Neural network Associative Classification system is implemented using Matlab 7.5.0 (R2007b). The experiments were performed on the Intel® Core i3 CPU, 2.27 GHz system running Windows 7 Home Basic with 2.00 GB RAM.

4.2 Methodology for generating the Class Association Rules set.

First, finding the Class Association Rules (CAR) from datasets can be useful in many contexts. In general understanding the CARs relates the class attribute with the predicted attributes. Extraction of association rules was achieved by using the proposed algorithm in algorithm 1. The rules are extracted depending on the relations between the predicted and class attributes. The dataset in the fig. 4 is a segment of an iris plant dataset.



```
sepal-length, sepal-width, petal-length, petal-width, class
6.7, 3.0, 5.2, 2.3, Iris-virginica.
6.0, 2.2, 5.0, 1.5, Iris-virginica.
6.2, 2.8, 4.8, 1.8, Iris-virginica.
7.7, 3.8, 6.7, 2.2, Iris-virginica.
7.2, 3.0, 5.8, 1.6, Iris-virginica.
5.5, 2.4, 3.8, 1.1, Iris-versicolor.
6.0, 2.7, 5.1, 1.6, Iris-versicolor.
5.5, 2.5, 4.0, 1.3, Iris-versicolor.
5.6, 2.9, 3.6, 1.3, Iris-versicolor.
5.7, 2.9, 4.2, 1.3, Iris-versicolor.
5.0, 3.2, 1.2, 0.2, Iris-setosa.
4.9, 3.1, 1.5, 0.1, Iris-setosa.
5.3, 3.7, 1.5, 0.2, Iris-setosa.
...
```

Fig. 4: An excerpt from an iris plant dataset

The aim is to use the proposed algorithm, to find the relations between the features and represents them in CARs to form the input to the backpropagation neural network. The three classes of Iris were allocated numeric representation as shown in table 4.

Table 4: Class and their numeric representation

Class	Numeric representation
Iris Virginica.	1
Iris Versicolour	2
Iris Setosa	3

Fig. 5 shows the snapshot of the main screen of our project. The import button will load the iris plant dataset into the MATLAB workspace. As the attributes are continuous, discretization is done.

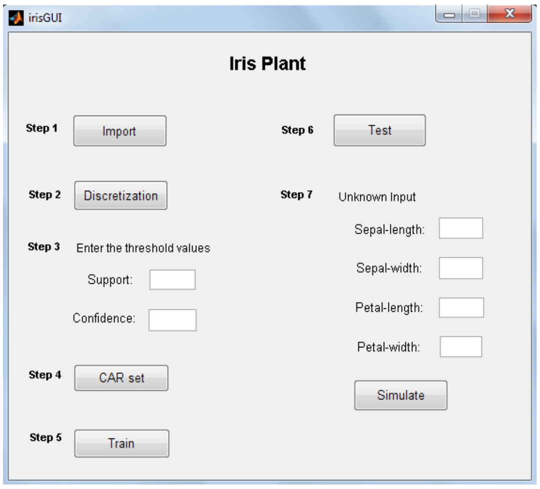
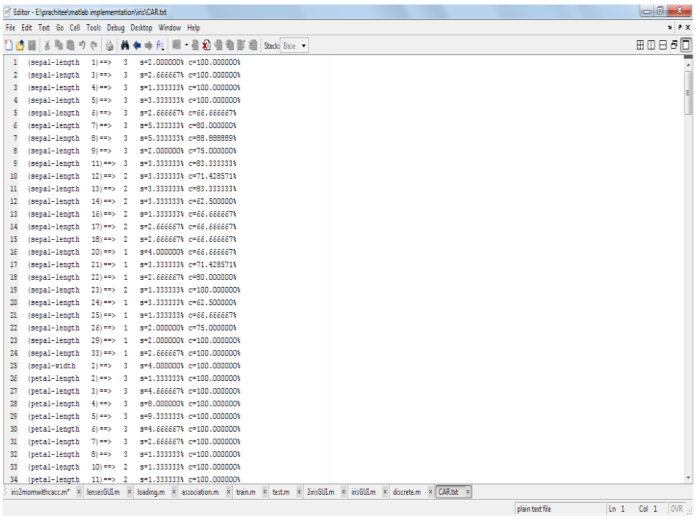


Fig.5: Neural Network Associative Classification System

With the user define support 1% and confidence 50%, CARs are extracted by using Algorithm 1and are shown in fig. 6, where s denotes support and c denotes confidence for that respective rule. Total 301 rules are generated from the iris plant dataset.



```
(sepal-length   1)==>  3   s=2.000000% c=100.000000%
(sepal-length   3)==>  3   s=2.666667% c=100.000000%
(sepal-length   4)==>  3   s=1.333333% c=100.000000%
(sepal-length   5)==>  3   s=3.333333% c=100.000000%
(sepal-length   6)==>  3   s=2.666667% c=66.666667%
(sepal-length   7)==>  3   s=5.333333% c=80.000000%
(sepal-length   8)==>  3   s=5.333333% c=88.888889%
(sepal-length   9)==>  3   s=2.000000% c=75.000000%
(sepal-length  11)==>  3   s=3.333333% c=83.333333%
(sepal-length  12)==>  2   s=3.333333% c=71.428571%
(sepal-length  13)==>  2   s=3.333333% c=83.333333%
(sepal-length  14)==>  2   s=3.333333% c=62.500000%
(sepal-length  16)==>  2   s=1.333333% c=66.666667%
(sepal-length  17)==>  2   s=2.666667% c=66.666667%
(sepal-length  18)==>  2   s=2.666667% c=66.666667%
(sepal-length  20)==>  1   s=4.000000% c=66.666667%
(sepal-length  21)==>  1   s=3.333333% c=71.428571%
(sepal-length  22)==>  1   s=2.666667% c=80.000000%
(sepal-length  23)==>  2   s=1.333333% c=100.000000%
(sepal-length  24)==>  1   s=3.333333% c=62.500000%
(sepal-length  25)==>  1   s=1.333333% c=66.666667%
(sepal-length  26)==>  1   s=2.000000% c=75.000000%
(sepal-length  29)==>  1   s=2.000000% c=100.000000%
(sepal-length  33)==>  1   s=2.666667% c=100.000000%
(sepal-width    2)==>  3   s=4.000000% c=100.000000%
(petal-length   2)==>  3   s=1.333333% c=100.000000%
(petal-length   3)==>  3   s=4.666667% c=100.000000%
(petal-length   4)==>  3   s=8.000000% c=100.000000%
(petal-length   5)==>  3   s=9.333333% c=100.000000%
(petal-length   6)==>  3   s=4.666667% c=100.000000%
(petal-length   7)==>  3   s=2.666667% c=100.000000%
(petal-length   8)==>  3   s=1.333333% c=100.000000%
(petal-length  10)==>  2   s=1.333333% c=100.000000%
(petal-length  11)==>  2   s=1.333333% c=100.000000%
```

Fig. 6: Generated Class Association Rules.

4.3 Methodology for building the classifier

For training the Backpropagation neural network CARs are used. To train the network, we have used the Matlab function train( ). In which 60% of data are used for training, 20% are used for testing and 20% for validation purpose. traingdm, learngdm and tansig are used as a training function, learning function and transfer function. Number of hidden layer nodes =9 and learning will stop when error is less than 0.005 or epochs= 5,000 times. These values are kept constant in order to determine momentum constant (mc) and learning rate (lr). The table 5 shows the number of misidentified testing patterns when momentum constant (mc) is varied and keeping learning rate constant to 0.9. The value for mc is selected for which misidentified testing patterns is minimum. The best performance is achieved when mc = 0.7.

Now, the values of learning rate are varied from 0.1 to 0.9 and mc is kept constant at 0.7 and numbers of misidentified testing patterns are shown in Table 6. So, best performance can be achieved at 0.2.

Table 5: Performance of Momentum Constant (mc)

Learning rate	**Momentum constant**	**Misidentified testing patterns**
0.9	0.1	29
	0.2	24
	0.3	23
	0.4	33
	0.5	26
	0.6	24
	0.7	7
	0.8	8
	0.9	19

Table 6: Performance of Learning Rate

Momentum constant	**Learning Rate**	**Misidentified testing patterns**
0.7	0.1	8
	0.2	2
	0.3	4
	0.4	3
	0.5	4
	0.6	3
	0.7	5
	0.8	9
	0.9	7

Now, to determine appropriate number of epochs, keep the values of all the other properties constant. Table 7 shows the number of misidentified patterns with varying epochs. The best performance is achieved with epochs = 2000 and Table 8 shows the best properties for Backpropagation neural network in order to build a classifier for iris plant.

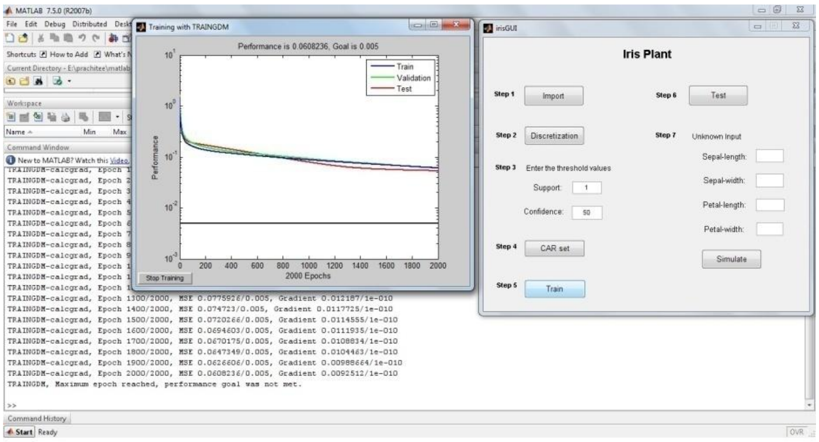
Table 7: Performance of Epochs

Epochs	Misidentified testing patterns
500	11
1000	6
1500	3
2000	2
2500	2
5000	2

Table 8: Properties to build Backpropagation neural network for Iris plant

Neural network Properties	Value
Number of hidden neurons	9
Transfer function	tansig
Learning rate	0.2
Momentum constant	0.7
Training technique	Gradient descent with momentum Backpropagation
Epochs	2000
Misidentified patterns	2
Training patterns	181
Validation patterns	60
Testing patterns	60

Fig. 7 and 8 shows the snapshot of the training and testing respectively with the properties shown in table 7.

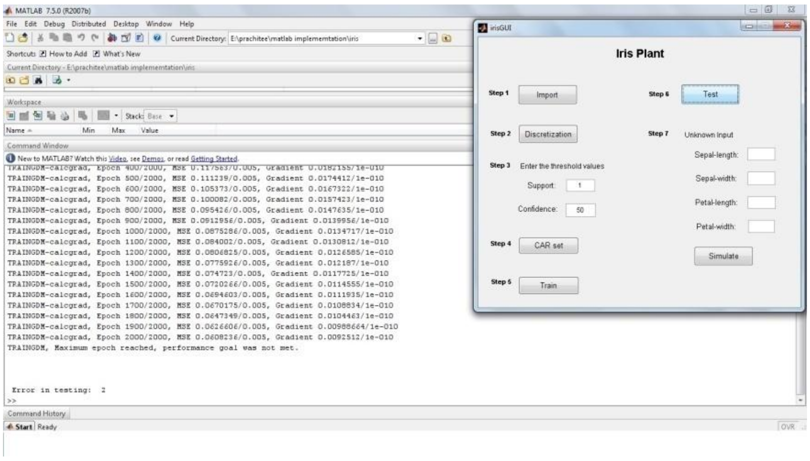


Support: 1
Confidence: 50

```
TRAINGDM-calcgrad, Epoch 1
TRAINGDM-calcgrad, Epoch 2
TRAINGDM-calcgrad, Epoch 3
TRAINGDM-calcgrad, Epoch 4
TRAINGDM-calcgrad, Epoch 5
TRAINGDM-calcgrad, Epoch 6
TRAINGDM-calcgrad, Epoch 7
TRAINGDM-calcgrad, Epoch 8
TRAINGDM-calcgrad, Epoch 9
TRAINGDM-calcgrad, Epoch 1
TRAINGDM-calcgrad, Epoch 1
TRAINGDM-calcgrad, Epoch 1
TRAINGDM-calcgrad, Epoch 1300/2000, MSE 0.0775926/0.005, Gradient 0.012187/1e-010
TRAINGDM-calcgrad, Epoch 1400/2000, MSE 0.074723/0.005, Gradient 0.0117725/1e-010
TRAINGDM-calcgrad, Epoch 1500/2000, MSE 0.0720266/0.005, Gradient 0.0114555/1e-010
TRAINGDM-calcgrad, Epoch 1600/2000, MSE 0.0694603/0.005, Gradient 0.0111935/1e-010
TRAINGDM-calcgrad, Epoch 1700/2000, MSE 0.0670175/0.005, Gradient 0.0108834/1e-010
TRAINGDM-calcgrad, Epoch 1800/2000, MSE 0.0647349/0.005, Gradient 0.0104463/1e-010
TRAINGDM-calcgrad, Epoch 1900/2000, MSE 0.0626606/0.005, Gradient 0.00988664/1e-010
TRAINGDM-calcgrad, Epoch 2000/2000, MSE 0.0608236/0.005, Gradient 0.0092512/1e-010
TRAINGDM, Maximum epoch reached, performance goal was not met.

>>
```

Fig. 7: Training the network


MATLAB 7.5.0 (R2007b)
File Edit Debug Distributed Desktop Window Help
Current Directory: E:\prachitee\matlab implememtation\iris
Shortcuts How to Add What's New
Current Directory - E:\prachitee\matlab implememtation\iris
Workspace
Stack: Base
Name Min Max Value
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
irisGUI
Iris Plant
Step 1
Import
Step 2
Discretization
Step 3
Enter the threshold values
Step 4
CAR set
Step 6
Train
Step 6
Test
Step 7
Unknown Input
Sepal-length:
Sepal-width:
Petal-length:
Petal-width:
Simulate
Command History
Start Ready
OVR

Support: 1
Confidence: 50

```
TRAINGDM-calcgrad, Epoch 400/2000, MSE 0.117563/0.005, Gradient 0.0182155/1e-010
TRAINGDM-calcgrad, Epoch 500/2000, MSE 0.111239/0.005, Gradient 0.0174412/1e-010
TRAINGDM-calcgrad, Epoch 600/2000, MSE 0.105373/0.005, Gradient 0.0167322/1e-010
TRAINGDM-calcgrad, Epoch 700/2000, MSE 0.100082/0.005, Gradient 0.0157423/1e-010
TRAINGDM-calcgrad, Epoch 800/2000, MSE 0.095426/0.005, Gradient 0.0147635/1e-010
TRAINGDM-calcgrad, Epoch 900/2000, MSE 0.0912956/0.005, Gradient 0.0139956/1e-010
TRAINGDM-calcgrad, Epoch 1000/2000, MSE 0.0875286/0.005, Gradient 0.0134717/1e-010
TRAINGDM-calcgrad, Epoch 1100/2000, MSE 0.084002/0.005, Gradient 0.0130812/1e-010
TRAINGDM-calcgrad, Epoch 1200/2000, MSE 0.0806825/0.005, Gradient 0.0126585/1e-010
TRAINGDM-calcgrad, Epoch 1300/2000, MSE 0.0775926/0.005, Gradient 0.012187/1e-010
TRAINGDM-calcgrad, Epoch 1400/2000, MSE 0.074723/0.005, Gradient 0.0117725/1e-010
TRAINGDM-calcgrad, Epoch 1500/2000, MSE 0.0720266/0.005, Gradient 0.0114555/1e-010
TRAINGDM-calcgrad, Epoch 1600/2000, MSE 0.0694603/0.005, Gradient 0.0111935/1e-010
TRAINGDM-calcgrad, Epoch 1700/2000, MSE 0.0670175/0.005, Gradient 0.0108834/1e-010
TRAINGDM-calcgrad, Epoch 1800/2000, MSE 0.0647349/0.005, Gradient 0.0104463/1e-010
TRAINGDM-calcgrad, Epoch 1900/2000, MSE 0.0626606/0.005, Gradient 0.00988664/1e-010
TRAINGDM-calcgrad, Epoch 2000/2000, MSE 0.0608236/0.005, Gradient 0.0092512/1e-010
TRAINGDM, Maximum epoch reached, performance goal was not met.



 Error in testing:  2
>>
```

Fig 8: Testing the network

Now we have a trained network, which can more accurately predict the class based on the predicted attributes.

It can be observed that extracted CARs include the most important features and informative relations from the dataset which will help neural network form the accurate classifier for iris plant.

4.4 Result

To assay the performance of classifier for iris plant using Neural Network Associative Classification system we will compare it with Classification Based Associations (CBA) on accuracy. Table 9 shows the accuracy of CBA and Neural network Associative Classification system on accuracy. Class association rules are obtained with minimum support= 1% and minimum confidence = 50%.

Table 9: Comparisons between the CBA system and the Neural Network Associative Classification system on accuracy

Dataset	CBA	Neural network Associative Classification system
Iris plant	94.7	96.667

From the experimental results, classifier for iris plant using Neural Network Associative Classification system will outperform than classifier for iris plant using CBA in accuracy because neural network will learn to adjust weight from the input that is class association rules and will build a more accurate and efficient classifier.

5. Conclusion

Neural network Associative Classification system is used for building accurate and efficient iris plant classifiers in order to improve its accuracy. The three-layer feed-forward back-propagation networks were developed by using Matlab. The optimal values for momentum constant, learning rate and epochs were found to be 0.7, 0.2 and 2000 respectively. The structure of networks reflects the knowledge uncovered in the previous discovery phase. The trained network is then used to classify unseen iris plant data.

Iris plant classifier using Neural Network Associative Classification system is more accurate than classifier build using CBA as shown in experimental result because weights are adjusted according to the class association rules and best one is used to classify the data.

In future work, we intend to apply predictive apriori algorithm for finding the class association rules instead of applying apriori algorithm.

References

[1]. U. Fayyad, G. Piatetsky-Shapiro, P. Smyth and R. Uthurusamy, “Advances in Knowledge Discovery and Data Mining”, MIT Press, Cambridge, Massachusetts, USA,1996.

[2]. Mary DeRosa, “Data Mining and data analysis for counterterrorism”, Center for Strategic and International Studies, March 2004.

[3]. Xindong wu,Vipin Kumar,et.al. “Top 10 algorithms in data mining”, Knowledge Information System(2008) 14:1-37 DOI 10.1007/s10115-007-0114-2,Springer-Verlag London Limited 2007.

[4]. R.Agrawal, T.Imieilinski, and A.Swami, “Mining Association Rules between sets of items in large databases”,SIGMOD,1993, pp. 207-216.

[5]. B.Liu, W. Hsu and Y.Ma, “Integrating Classification and Association rule mining”, KDD,1998, pp. no.80-86.

[6]. Jiawei Han and Micheline Kamber, “Data Mining Concepts and Techniques”, 2e Elsevier Publication, 2008.

[7]. Robert J. Schalkoff, “ Artificial Neural networks”, McGraw Hill publication, ISE.

[8]. Chung, Kusiak, “Grouping parts with a neural network”, Journal of Manufacturing System, volume 13, Issue 4, 02786125, April 2003, pp.no. 262.

[9]. Prachitee B. Shekhawat, Sheetal S. Dhande, “A classification technique using associative classification”, International journal of computer application(0975-8887) vol. 20-No.5,April 2011,pp.no. 20-28.

[10]. Cheng Jung Tsai, Chein-I Lee, Wei-Pang Yang, “A discretization algorithm based on Class Attribute Contingency Coefficient”, Inf. Sci., 178(3),2008.pp. no. 714-731.

[11]. http://archive.ics.uci.edu/ml/datasets.html.

[12]. Xiaoxin Yin,Jiawei Han, “CPAR: Classification based on Predictive Association Rules”, In Proc. Of SDM, 2003, pp.no. 331-335.