Knowledge Discovery by Hopfield Network
ホップフィールドニューラルネットワークによる
定性推論知識発見

Thanakorn SORNKAEW and Yasuo YAMASHITA
ソンケオ タナコン*†, 山下 安雄*

In this paper, a new method for inducing symbolic knowledge from empirical data is proposed. This method consists of four steps. Firstly, linguistic variables of fuzzy membership functions are simply defined using an analysis of histograms. Secondly, weights of the Hopfield network are calculated. Thirdly, the states of the Hopfield network are asynchronously updated until these states remain unchanged. In the fourth step, rules are extracted via our proposed algorithm. The experiments in three data sets show good performance.

本論文では、ポップフィールドニューラルネットワークを用いて経験的データから定性推論知識を発見する手法を提案する。本手法は、4つのステップから成り立つ。まず、ヒストグラムの分析により、ファジー・メンバーシップ関数の言語的変数を定義する。次に、ポップフィールドニューラルネットワークの重みを計算する。そして、ポップフィールドニューラルネットワークの状態が安定になるまで同期的におかれる。最後に、提案する抽出アルゴリズムにより、定性推論知識を抽出することができる。本手法の有効性を3種類のデータを用いた実験により確認した。

Keywords: Hopfield network, symbolic knowledge, membership function, fuzzy, rule extraction
ポップフィールドニューラルネットワーク, 定性推論知識, メンバーシップ関数, ファジー, ルール抽出

1 Introduction
The knowledge discovery draws on findings from statistics and/or databases to construct tools that allow insight into massive data sets. An important goal of knowledge discovery is to turn data into knowledge. A framework for knowledge discovery processes described in [1] is developing an understanding of the application domain, creating a target data set and focusing on a subset of variables, cleaning and preprocessing data, reducing and transforming data, choosing the data mining task, mining data, evaluating the output, and consolidating discovered data. The knowledge discovery is usually represented in rule formats. Knowledge based neural networks are widely used as tools for discovering knowledge and in rule extraction from data sets.

Most of the rules extraction and knowledge discovery methods have been proposed based on MLPs [3-8]. Many researchers have achieved high accuracy rates with the MLP models. Logical rules extraction has also been attempted using a self-organizing map [9], ART [11], and fuzzy ART architecture [15]. Simple self-organizing architectures may also be used for rule extraction [17], although accuracy of the self-organizing mapping for classification
problems is rather poor.

The Hopfield network was introduced by Hopfield and Tank in 1985 [28], [29] for optimization problems. Since then the neural networks have proved effective in dealing with many combinatorial optimization problems [12], [13], [19], [20]. The Hopfield networks are easily trained in single pass [21]. The Hopfield network is classified as a fixed-weight neural network model as opposed to a supervised or an unsupervised model, where the networks are trained adaptively using a learning algorithm. The Hopfield network also shows the most efficient use of the neurons and connections as in the Boltzmann machine [2]. Thus the advantages of using the Hopfield neural network for knowledge discovery are the speedy and easy learning of the connection weights and the reduction of the number of neurons and connection weights which influence the number of rules. The Hopfield networks do not appear in knowledge discovery and rule extraction fields.

This paper introduces a new method for knowledge discovery using the Hopfield network and also proposes a rule extraction algorithm based on an analytic approach for inducing symbolic knowledge from empirical data. In Section 2, we describe an overview of the proposed method, fuzzy rules, fuzzy linguistic variables, and the Hopfield network. In Section 3 a proposed rule extraction algorithm is described and experimental results are presented in Section 4.

2 Knowledge Discovery Using the Hopfield Network

This method is based on the computing paradigm of the Hopfield networks. Fig. 1 presents the major steps in our proposed method based on the Hopfield network for extracting rules from data sets. Firstly, we begin with finding the fuzzy linguistic variables by dividing the range of each feature into two or three parts depending on the data set through the analysis of histograms. Then we use these variables to determine initial states of the Hopfield network. Secondly, the Hopfield network’s connection weights are easily calculated via Hebb’s postulate of learning. Thirdly, the network states are updated until convergence then the output of the network is derived at the end. Finally, the rules are extracted using our proposed rule extraction algorithm. The details of our proposed method are described in the following.
2.1 Defining fuzzy rules and fuzzy linguistic variables

We consider a multi-input single-output fuzzy model based on the collection of rules which was developed in [18]. This knowledge representation in fuzzy modeling is presented in the following format:

\[
\text{IF } X_1 \text{ is } A_1 \text{ AND } X_2 \text{ is } A_2 \text{ AND}, \ldots, \text{AND } X_n \text{ is } A_n \text{ THEN Class is } V_q.
\]

We used this rule format for the following advantages. The consequent parts are presented in linguistic terms which make this fuzzy rule format more intuitive and understandable. Also, this rule format is easy to implement [20].

The flexibility of the fuzzy approach depends on the choice of membership functions. Fuzzy classifiers usually use a few membership functions per input feature [23]. We used context dependent linguistic variables and triangular membership functions for this work. Triangular membership functions may be regarded as a piece-wise linear approximation to Gaussian membership functions. For real-valued attributes, intervals are needed for defining linguistic variables used in logical rules. These intervals are determined by analysis of histograms, self-organizing feature map neural network [7], [16], decision tree [10], special “linguistic units” in multilayer perceptron network [14]. We used the analysis of histograms displaying the data of each feature in selecting linguistic variables. This method was suggested and used by many researchers [23], [27]. Histograms should be smoothed, by assuming that each data is really a triangular fuzzy member.

2.2 The Hopfield network for modeling knowledge

The Hopfield network can be viewed as nonlinear associative memory [22]. The primary function of this associative memory is to retrieve a pattern which is stored in the memory in response to the presentation of a noisy pattern. Moreover, this memory is error-correcting in the sense that it can override inconsistent information. Basically, the starting state is close to the stable state which is represented in memory. Then the network evolves with time until convergence.

In Fig. 2, the circles show neurons of the Hopfield network and the black dots describe the network’s connection weights. For a network made up of neurons, the state of the network is thus defined by the potential of neurons,

\[
S = [S_{1,1}, S_{1,2}, \ldots, S_{1,K}, S_{2,1}, S_{2,2}, \ldots, S_{2,K}, \ldots, S_{M,1}, S_{M,2}, \ldots, S_{M,K}]^T
\]

where the superscript \( T \) denotes matrix transposition, \( M \) is the total number of attributes, and \( K \) is the number of fuzzy membership functions. Initialization of state \( S_{i,k} \) at time \( t = 0 \) is determined by converting attribute value to fuzzy value:

\[
S_{i,k}(0) = f_{i,k}(x_i)
\]

where \( f_{i,k}(\cdot) \) is the \( k \) th fuzzy membership function for the \( i \) th attribute, and \( x_i \) is the \( i \) th attribute value.

A pair of neurons \( ik \) and \( jl \) in the network is connected by a connection weight \( W_{ik,jl} \) which specifies the contribution of the output \( Y_{j,l} \) of neuron \( jl \) to the potential acting on neuron \( ik \). In any iteration, the network states \( S_{i,k} \) is determined by [31]

\[
\frac{dS_{i,k}}{dt} = \sum_{j=1}^{M} \sum_{l=1}^{K} W_{ik,jl} Y_{j,l} + H_{i,k}
\]
where $H_{i,k}$ is the threshold of the $i$ th attribute at the $k$ th fuzzy membership function. According to the above equation, the network states are gradually changed with time. The output of neuron $Y_{i,k}$ depends on the value of the network state $S_{i,k}$ according to the sigmoid function:

$$Y_{i,k} = \frac{1}{1 + \exp(-S_{i,k}/T_e)}$$  \hspace{1cm} (4)

where $T_e$ is a pseudotemperature and $Y_{i,k} \in [0,1]$. In order to get the expression of the network weights, we need to define an energy function which is to be minimized. The energy function of the continuous version of the Hopfield network considered here is defined by [31]

$$E = -\frac{1}{2} \sum_{i=1}^{M} \sum_{k=1}^{K} \sum_{j=1}^{M} \sum_{l=1}^{K} W_{ik,jl} Y_{i,k} Y_{j,l}$$

$$- \sum_{i=1}^{M} \sum_{k=1}^{K} H_{i,k} Y_{i,k}.$$  \hspace{1cm} (5)

State changes will continue until a local minimum of the energy landscape is reached where $dY_{i,k}/dt = 0$: that is, to zero, one, or to somewhere which satisfies $\partial E/\partial Y_{i,k} = 0$. At equilibrium, neuron $ik$ takes value 0, 1, or a value between zero and one.

According to the generalization of Hebb's postulate of learning, the synaptic weight from neuron $jl$ to neuron $ik$ is calculated by

$$W_{ik,jl} = \frac{1}{N} \sum_{\mu=1}^{P} \xi_{\mu,ik} \xi_{\mu,jl} - \frac{P}{N} I$$  \hspace{1cm} (6)

where $N$ is the total number of neurons, $P$ is the total number of teaching patterns, $\xi_{\mu,ik}$ is fuzzy value of the $i$ th attribute of teaching pattern $\mu$ at the $k$ th fuzzy membership function, and $I$ denotes the identity matrix.

There are three conditions for synaptic weight $W_{ik,jl}$.

1. The output of each neuron in the network is fed back to all other neurons. Also, the states of neurons are asynchronously updated at any iteration.
2. The influence of neuron $ik$ on neuron $jl$ is equal to the influence of neuron $jl$ on neuron $ik$.

3. For every neuron, there is no self-feedback.

The Hopfield network is known to have a limitation in the number of patterns that can be stored and accurately recalled. Hopfield [29], for example, has shown that the number of classes stored should be kept well below 0.15 times the number of neurons in the network. Therefore, we used the Hopfield network to model knowledge for each class.

3 Proposed Rule Extraction Algorithm

The basic idea of our proposed rule extraction algorithm is to analyze how the Hopfield network operates and recognizes input patterns. As we described in the previous section, the Hebb's postulate of learning uses the multiplication of fuzzy values shown in Eq. (6) to construct network's connection weights, therefore, the accuracy of pattern recognition is dependent on converting attribute values to network's fuzzy values. The Hopfield network retrieves the memory and the network state is updated until a local minimum of the energy landscape is reached. The stable states correspond to the network's initial states which are nominally assigned by the memory of the network. The stable states are differently located according to the classes.

Each attribute value is converted to a few initial states of the network through fuzzy membership functions. Mostly, one fuzzy membership function matches the input attribute value and results in large initial state. Therefore, after the network has converged, there is one large stable state in each attribute. The large stable states can be used for inducing rules as hidden neurons of MLPs. The reason why we choose only large stable states for generating rules is that the connection weights, which are linked from neurons that hold large initial states to neurons that hold large stable states, are larger than others.

Basically, the network states are influenced by the connection weights as shown in Eq. (3) so that the larger the value of the connection weight, the more important the attribute to the network state. Generally, we can correctly discriminate input patterns by using some distinguished attributes. Consequently, we used these ideas to develop the algorithm for rule extraction. The details of the algorithm are described below.

The proposed rule extraction algorithm consists of five procedures. Firstly, since the learning in different classes gives different output, the sum of the output for all teaching patterns is calculated. Secondly, the maximum sum of the output in each attribute is selected as one of candidate rules. Thirdly, for simple understanding, all attributes do not need to be antecedents of extracted rules so that the more important attributes are used. The important attributes and linguistic variables are determined by the connection weights linked to the neuron which is selected as a candidate rule. Fourthly, rules are generated in the selected format. Finally, the same rules are merged. The algorithm proceeds separately for each class. The fuzzy rules for each class can be discovered through the following procedures:

1) For each attribute \(i\) (\(i = 1 \text{ to } M\)), for each fuzzy membership function \(k\) (\(k = 1 \text{ to } K\)),

- compute the sum of output \(Y_{i,k}\) derived from trained neural network
for all teaching patterns \( \mu \),
\[
O_{i,k} = \sum_{\mu=1}^{P} Y_{i,k}^\mu (\text{stable}). \quad (7)
\]

2) The following algorithm is operated in order to determine the candidate rules. For each attribute \( i \) \((i = 1 \text{ to } M)\), select the maximum sum of outputs \( O_{i,k} \), then put neuron number \( i k \) into rule list \( Q = \{ i k_1 \ldots i k_q \} \).

3) Select \( F \) number of connection weights to infer attributes and fuzzy values through the use of the proposed algorithm shown below:

For each candidate rule \( i k \) in list \( Q = \{ i k_1 \ldots i k_q \} \),

- for each \( j \) \((j = 1 \text{ to } M) \) and \( j \neq i \), for \( l = 1 \text{ to } K \), select the maximum connection weight \( W_{i k,j} \) linked to
  - sort all maximum connection weights from large to small value.
  - select \( F \) number of connection weights which are sorted in large value.
  - thus, we can define the \( f \) th antecedent fuzzy value \( A_f \) equal to the \( l \) th fuzzy linguistic term and the \( f \) th antecedent input value \( X_f \) equal to the \( j \) th attribute.

4) For each candidate fuzzy rule in rule list \( Q \), our algorithm generates fuzzy rules in the format:
\[
R: \text{IF} \quad X_1 = A_1 \land X_2 = A_2 \land \ldots, \quad X_f = A_f \land \ldots, \quad X_F = A_F \quad \text{THEN} \quad \text{Class} = V \quad (8)
\]

where \( R \) is the fuzzy rule in rule list \( Q \), \( X_f \) is input value of the \( f \) th antecedent, \( A_f \) is fuzzy variable of the \( f \) th antecedent, and \( V \) is a class type of the fuzzy rule in the consequent part.

5) Merge the same rules and decrease the number of fuzzy rules.
For each candidate rule \( i \) in list \( Q = \{ i k_1 \ldots i k_q \} \), for each candidate rule \( j \) in list \( Q = \{ i k_1 \ldots i k_q \} \) and \( j \neq i \),

- for each antecedent \( f \), search for \( X_f \) and \( A_f \) which match.

Delete the same fuzzy rules and update the number of fuzzy rules.

Defining the number of antecedents, \( F \), is dependent on the learning task. When the data set is not complex and obviously categorized, the number of antecedents equal to 0.2–0.3 of the total number of attributes should be selected to be easily understood. In other words, when the data set is complicated the number of antecedents equal to 0.3–0.4 of the total number of attributes should be set to avoid misclassification. However, the number of antecedents, \( F \), should not be defined below 2.

4 Experimental Results

We used three data sets, “Iris plants data set”, “Wisconsin breast cancer data set”, and “Cleveland heart disease data set”, from machine learning databases of UCI [26] for this study. We set thresholds \( H_{ik} \) to zero, \( T_e \) equal to 0.8 for all experiments, and \( F \) equal to 2–4.
4.1 Experiments on the Iris plants recognition data set

This data set contains 3 classes of 50 instances each. The types of iris plants are setosa, versicolor, and virginica. Each instance consists of 4 attributes which are sepal length $x_1$, sepal width $x_2$, petal length $x_3$, and petal width $x_4$. All attribute values are continuous and given in centimeters. The simplest way to obtain linguistic variables is based on the division of each attribute data range into a number of parts and the use of triangular membership functions for each part [23]. Cutting the histograms into the regions where values of attributes are most frequently found around the center of the regions in a given class provides almost optimal division of fuzzy linguistic variables.

For example, Iris-setosa are more frequent for the value $x_1$ below 5.5 and Iris-versicolor are more frequent above this value. Fig. 3 and Table 1 show histograms of four attributes based on the Iris plants data and all linguistic variables of attributes which were obtained by the analysis of histograms and which were modified from data in [27]. We divided each attribute of the Iris plants into three parts: small (s), medium (m), large (l), according to the characteristics of the histograms. The attribute $x_1$ is called small if it is in the [4.2, 5.5] range, medium in the [5.0, 6.1], and large in the [5.7, 8.0]. Then, the attribute values are converted to fuzzy values in 0–1.0 range through the triangular membership functions which are constructed using data intervals of linguistic variables. For instance, triangular membership func-

Fig. 3 Histograms of the attribute $x_1 - x_4$ of Iris plants.
Table 1 Linguistic variables of the Iris plants data set obtained by analysis of histogram.

<table>
<thead>
<tr>
<th></th>
<th>Small(s)</th>
<th>Medium(m)</th>
<th>Large(l)</th>
</tr>
</thead>
<tbody>
<tr>
<td>x1</td>
<td>[4.2, 5.5]</td>
<td>[5.0, 6.1]</td>
<td>[5.7, 8.0]</td>
</tr>
<tr>
<td>x2</td>
<td>[1.8, 2.7]</td>
<td>[2.5, 3.5]</td>
<td>[3.0, 4.4]</td>
</tr>
<tr>
<td>x3</td>
<td>[1.0, 2.05]</td>
<td>[2.85, 5.2]</td>
<td>[4.4, 7.1]</td>
</tr>
<tr>
<td>x4</td>
<td>[0, 0.6]</td>
<td>[0.8, 1.8]</td>
<td>[1.4, 2.7]</td>
</tr>
</tbody>
</table>

Fig. 4 Triangular membership functions of the attribute $x_1$ of Iris plants.

ections of $x_1$ are constructed as Fig. 4. The peak of each triangular membership function is located in the center of each data interval. We can also present the triangular membership functions of $x_1$ as

$$f_{1,1}(x_1) = \begin{cases} 
  1.54 \times (x_1 - 4.2) & 4.2 \leq x_1 \leq 4.85 \\
  -1.54 \times (x_1 - 5.5) & 4.85 \leq x_1 \leq 5.5 \\
  0 & \text{otherwise}
\end{cases}$$

$$f_{1,2}(x_1) = \begin{cases} 
  1.82 \times (x_1 - 5.0) & 5.0 \leq x_1 \leq 5.55 \\
  -1.82 \times (x_1 - 6.1) & 5.55 \leq x_1 \leq 6.1 \\
  0 & \text{otherwise}
\end{cases}$$

$$f_{1,3}(x_1) = \begin{cases} 
  0.87 \times (x_1 - 5.7) & 5.7 \leq x_1 \leq 6.85 \\
  -0.87 \times (x_1 - 8.0) & 6.85 \leq x_1 \leq 8.0 \\
  0 & \text{otherwise}
\end{cases}$$

Cross-validation is important to evaluate the ability of learning domain rules if they are not available. Domain validity is indicated if rules learned through some data can be well applied to other data. We used the same neural network structure which consisted of 12 neurons in three learning classes. The extracted rules for each category are shown below:

**Extracted Rules for Setosa**

$R_1$: if $x_1 = s \land x_3 = s$ then class = setosa

$R_2$: if $x_1 = s \land x_4 = s$ then class = setosa

$R_3$: if $x_3 = s \land x_4 = s$ then class = setosa

**Extracted Rules for Versicolor**

$R_1$: if $x_2 = m \land x_3 = m$

then class = versicolor

$R_2$: if $x_2 = m \land x_4 = m$

then class = versicolor

$R_3$: if $x_3 = m \land x_4 = m$

then class = versicolor

**Extracted Rules for Virginiga**

$R_1$: if $x_1 = l \land x_3 = l$

then class = virginiga

$R_2$: if $x_2 = m \land x_3 = l$

then class = virginiga

$R_3$: if $x_3 = l \land x_4 = l$

then class = virginiga.

Performance is evaluated by the accuracy rate, simply, the ratio of the number of correctly classified test examples to the total number of test examples. The experimental results of the 10-fold cross-validation showed...
good performance in 98.0 percent of the total number of examples. We also compared our proposed method to other works and the resulting comparison is shown in Table 2. Our proposed method shows higher accuracy than all neurofuzzy systems, ReFuNN [23], FuNe-I [25], and NEFCLASS [24].

4.2 Experiments on the Wisconsin breast cancer data set

The Wisconsin breast cancer data set contains 699 instances: 458 benign and 241 malignant. In each of the instances, there are nine attributes with integer value 1 to 10, binary class label and sample code number. Attributes of each instance consist of clump thickness $x_1$, uniformity of cell size $x_2$, uniformity of cell shape $x_3$, marginal adhesion $x_4$, single epithelial cell size $x_5$, bare nuclei $x_6$, bland chromatin $x_7$, normal nucleoli $x_8$, and mitoses $x_9$. For 16 instances one attribute is missing, so we deleted attributes with missing values. This data has been analyzed in a number of papers. Fig. 5 shows the histograms of the nine attributes from the Wisconsin breast cancer data. Using the analysis of the histograms, the range of each feature is divided into two parts, small (s) and large (l) as shown in Table 3. These are data-driven fuzzy linguistic variables. Attribute values are, then, converted to fuzzy values through the triangular membership functions. The network structure consists of 18 neurons. The total number of extracted rules is 16, 7 extracted rules for class benign and 9 extracted rules for class malignant. All extracted rules of each class are shown below:

**Extracted Rules for Benign**

$R_1$: if $x_2 = s \land x_3 = s \land x_4 = s$ then class = benign

$R_2$: if $x_3 = s \land x_6 = s \land x_8 = s$ then class = benign

$R_3$: if $x_6 = s \land x_8 = s \land x_9 = s$ then class = benign

$R_4$: if $x_6 = s \land x_7 = s \land x_9 = s$ then class = benign

$R_5$: if $x_4 = s \land x_6 = s \land x_8 = s$ then class = benign

$R_6$: if $x_2 = s \land x_8 = s \land x_9 = s$ then class = benign

$R_7$: if $x_2 = s \land x_7 = s \land x_9 = s$ then class = benign

**Extracted Rules for Malignant**

$R_1$: if $x_2 = l \land x_3 = l \land x_5 = l$ then class = malignant

$R_2$: if $x_3 = l \land x_5 = l \land x_6 = l$ then class = malignant

$R_3$: if $x_5 = l \land x_6 = l \land x_9 = s$ then class = malignant

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy (%)</th>
<th>The number of rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>FuNe-I [25]</td>
<td>96.0</td>
<td>7</td>
</tr>
<tr>
<td>NEFCLASS [24]</td>
<td>96.7</td>
<td>7</td>
</tr>
<tr>
<td>ReFuNN [23]</td>
<td>95.7</td>
<td>104</td>
</tr>
<tr>
<td>Proposed Method</td>
<td>98.0</td>
<td>9</td>
</tr>
</tbody>
</table>

*Table 2 Results from the 10-fold cross-validation for the Iris plants data set.*
$R_4$: if $x_5 = l \land x_6 = l \land x_7 = l$
then class = malignant

$R_5$: if $x_6 = l \land x_8 = s \land x_9 = l$
then class = malignant

$R_6$: if $x_1 = s \land x_5 = s \land x_9 = l$
then class = malignant

$R_7$: if $x_1 = s \land x_3 = s \land x_9 = l$
then class = malignant

$R_8$: if $x_1 = l \land x_3 = l \land x_6 = l$
then class = malignant

$R_9$: if $x_1 = l \land x_7 = l \land x_8 = l$
then class = malignant.

The results from the 10-fold cross-validation for the Wisconsin breast cancer data set achieved 97.1 percent overall accuracy. This result is comparable with other

---

**Fig. 5** Histograms of the attribute $x_1$-$x_9$ of Wisconsin breast cancer.
Table 3  Linguistic variables of the Wisconsin breast cancer data set obtained by analysis of histograms.

<table>
<thead>
<tr>
<th></th>
<th>Small (s)</th>
<th>Large (l)</th>
</tr>
</thead>
<tbody>
<tr>
<td>x1</td>
<td>[-1.1, 6.1]</td>
<td>[3.5, 12.0]</td>
</tr>
<tr>
<td>x2</td>
<td>[-1.3, 3.3]</td>
<td>[1.8, 11.5]</td>
</tr>
<tr>
<td>x3</td>
<td>[-1.3, 3.3]</td>
<td>[1.8, 11.6]</td>
</tr>
<tr>
<td>x4</td>
<td>[-1.3, 3.3]</td>
<td>[1.7, 12.0]</td>
</tr>
<tr>
<td>x5</td>
<td>[0.7, 3.2]</td>
<td>[1.8, 11.2]</td>
</tr>
<tr>
<td>x6</td>
<td>[-1.3, 3.3]</td>
<td>[1.7, 12.5]</td>
</tr>
<tr>
<td>x7</td>
<td>[-0.4, 4.1]</td>
<td>[2.5, 11.0]</td>
</tr>
<tr>
<td>x8</td>
<td>[-1.3, 3.3]</td>
<td>[1.9, 11.2]</td>
</tr>
<tr>
<td>x9</td>
<td>[-0.2, 2.2]</td>
<td>[0, 10.5]</td>
</tr>
</tbody>
</table>
methods as shown in Table 4. Our results show higher accuracy than the methods of Naive Bayes, MLP+backprop, CART(dec. tree), LVQ [30], and NEFCLASS [24].

4.3 Experiments on the Cleveland heart disease data set

The Cleveland heart disease data set, which was collected at the V.A Medical Center, Long Beach and the Cleveland Clinic Foundation, contains 303 instances. 164 healthy instances with the remaining being heart disease instances. While the database has 76 raw attributes, only 14 are actually used that include 9 continuous values, 4 nominal values and one class label. The attributes are age \(x_1\), sex \(x_2\), chest pain type \(x_3\), resting blood pressure \(x_4\), serum cholesterol in mg/dl \(x_5\), fasting blood sugar \(x_6\), resting electrocardiographic results \(x_7\), maximum heart rate achieved \(x_8\), exercise induced angina \(x_9\), ST depression included by exercise relative to rest \(x_{10}\), the slope of the peak exercise ST segment \(x_{11}\), the number of major vessels colored by flurosopy \(x_{12}\), and thal \(x_{13}\). There are many missing values in the data set so that we deleted those attributes to avoid the missing value problem. This data set is difficult to correctly classify but is used by many researchers. The histograms of thirteen attributes from the Cleveland heart disease data are shown in Fig. 6. The linguistic variables are driven by the data which is devided into 2–3 parts, low(l), middle(m) and high(h) as shown in Table 5. Attribute values are, then, coded into fuzzy values through the use of the triangular membership functions. Therefore, the network structure consists of 32 neurons. All 24 extracted rules are shown below:

**Extracted Rules for Healthy**

\[R_1: \text{if } x_5 = l \land x_8 = h \land x_{10} = l \land x_{11} = l \text{ then class = healthy} \]
\[R_2: \text{if } x_1 = l \land x_2 = h \land x_9 = l \land x_{10} = l \text{ then class = healthy} \]
\[R_3: \text{if } x_1 = l \land x_{10} = l \land x_{12} = l \land x_{13} = l \text{ then class = healthy} \]
\[R_4: \text{if } x_2 = h \land x_5 = l \land x_8 = h \land x_{10} = l \text{ then class = healthy} \]
\[R_5: \text{if } x_1 = l \land x_5 = l \land x_8 = h \land x_{12} = l \text{ then class = healthy} \]
\[R_6: \text{if } x_5 = l \land x_9 = l \land x_{10} = l \land x_{12} = l \text{ then class = healthy} \]
\[R_7: \text{if } x_2 = h \land x_9 = l \land x_{10} = l \land x_{12} = l \text{ then class = healthy} \]
\[R_8: \text{if } x_4 = l \land x_9 = l \land x_{11} = l \land x_{13} = m \text{ then class = healthy} \]
\[R_9: \text{if } x_{10} = l \land x_{11} = l \land x_{12} = l \land x_{13} = l \text{ then class = healthy} \]
Extracted Rules for Heart Disease

R1: if \( x_2 = h \land x_9 = h \land x_{10} = l \land x_{11} = m \land x_{12} = l \) then class = heart disease

R2: if \( x_4 = l \land x_6 = l \land x_{11} = m \land x_{13} = h \) then class = heart disease

R3: if \( x_2 = h \land x_7 = h \land x_8 = m \land x_{12} = l \) then class = heart disease

R4: if \( x_1 = l \land x_2 = h \land x_5 = l \land x_6 = l \) then class = heart disease

R5: if \( x_2 = l \land x_3 = l \land x_7 = h \land x_{12} = l \) then class = heart disease

R6: if \( x_5 = l \land x_{11} = m \land x_{12} = l \land x_{13} = h \) then class = heart disease

R7: if \( x_4 = h \land x_5 = h \land x_7 = h \land x_9 = l \) then class = heart disease
Fig. 6 (Continued.)
Table 5  Linguistic variables of the Cleveland heart disease data set obtained by analysis of histograms.

<table>
<thead>
<tr>
<th></th>
<th>Low (l)</th>
<th>Middle (m)</th>
<th>High (h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>x1</td>
<td>[32.0, 59.5]</td>
<td></td>
<td>[49.5, 78.0]</td>
</tr>
<tr>
<td>x2</td>
<td>[0]</td>
<td></td>
<td>[1.0]</td>
</tr>
<tr>
<td>x3</td>
<td>[0.4, 2]</td>
<td></td>
<td>[1.8, 5.8]</td>
</tr>
<tr>
<td>x4</td>
<td>[98.0, 132.5]</td>
<td>[123.5, 148.0]</td>
<td>[140.5, 193.0]</td>
</tr>
<tr>
<td>x5</td>
<td>[128.0, 256.5]</td>
<td></td>
<td>[242.5, 363.0]</td>
</tr>
<tr>
<td>x6</td>
<td>[0]</td>
<td></td>
<td>[1.0]</td>
</tr>
<tr>
<td>x7</td>
<td>[0]</td>
<td>[1.0]</td>
<td>[2.0]</td>
</tr>
<tr>
<td>x8</td>
<td>[94.0, 155.0]</td>
<td>[134.0, 166.0]</td>
<td>[158.0, 193.0]</td>
</tr>
<tr>
<td>x9</td>
<td>[0]</td>
<td></td>
<td>[1.0]</td>
</tr>
<tr>
<td>x10</td>
<td>[−1.0, 1.2]</td>
<td>[0.75, 2.65]</td>
<td>[2.3, 6.2]</td>
</tr>
<tr>
<td>x11</td>
<td>[1.0]</td>
<td>[2.0]</td>
<td>[3.0]</td>
</tr>
<tr>
<td>x12</td>
<td>[−1.1, 1.25]</td>
<td></td>
<td>[−0.7, 3.4]</td>
</tr>
<tr>
<td>x13</td>
<td>[3.0]</td>
<td>[6.0]</td>
<td>[7.0]</td>
</tr>
</tbody>
</table>

Table 6  Results from the 10-fold cross-validation for the Cleveland heart disease data set.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naive Bayes [30]</td>
<td>83.4</td>
</tr>
<tr>
<td>CART (decision tree) [30]</td>
<td>80.8</td>
</tr>
<tr>
<td>LVQ [30]</td>
<td>82.9</td>
</tr>
<tr>
<td>MLP+backprop [30]</td>
<td>81.3</td>
</tr>
<tr>
<td>Proposed method</td>
<td>83.8</td>
</tr>
</tbody>
</table>
5 Conclusions

In this paper, we have proposed a new method which is based on the Hopfield network and which can extract fuzzy rules or discover knowledge from empirical data. We firstly determined the linguistic variables of fuzzy membership functions through the analysis of histograms. Initial states of the Hopfield network are derived by converting raw input data through triangular membership functions. The connection weights are calculated using the Hebb’s postulate of learning and then the network states are updated until convergence. A rule extraction algorithm for discovering knowledge or extracting symbolic rules is proposed. The proposed algorithm is based on an analytical approach. We used Iris plants, Wisconsin breast cancer, and Cleveland heart disease databases to evaluate our proposed technique. According to our experiments, our method is able to extract meaningful rules for all three data sets. The discovered rules had good performance with high accuracy rates and are comparable to other methods. We realize that the analysis of histogram is subjective and we intend to improve on defining membership functions through using other techniques, e.g. the Gaussians mixture.

References


(Received March 16, 2004)
(Accepted September 6, 2004)