Framewise Phone Classification using Weighted Fuzzy Classification Rules

Omid Dehzangi\textsuperscript{1}, Bin Ma\textsuperscript{2}, Eng Siong Chng\textsuperscript{1}, Haizhou Li\textsuperscript{1,2,3}

\textsuperscript{1} School of Computer Engineering, Nanyang Technological University, Singapore
\textsuperscript{2} Institute for Infocomm Research, A*STAR, Singapore 138632
\textsuperscript{3} Department of Computer Science and Statistics, University of Eastern Finland, FI-80101 Joensuu, Finland
{omid0002, aseschng}@ntu.edu.sg, {mabin, hli}@i2r.a-star.edu.sg

Abstract

Our aim in this paper is to propose a rule-weight learning algorithm in fuzzy rule-based classifiers. The proposed algorithm is presented in two modes: first, all training examples are assumed to be equally important and the algorithm attempts to minimize the error-rate of the classifier on the training data by adjusting the weight of each fuzzy rule in the rule-base, and second, a weight is assigned to each training example as the cost of misclassification of it using the class distribution of its neighbors. Then, instead of minimizing the error-rate, the learning algorithm is modified to minimize the sum of costs for misclassified examples. Using six data sets from UCI-ML repository and the TIMIT speech corpus for framewise phone classification, we show that our proposed algorithm considerably improves the prediction ability of the classifier.

1. Introduction

Fuzzy systems have been widely used to solve control problems [1]. Recently, fuzzy rule-based systems have been applied to classification problems [2]. The interest in using fuzzy rule-based systems for classification problems arises from the fact that these systems are highly interpretable, which makes them a suitable tool for knowledge representation.

The main challenge in designing fuzzy classification systems is to generate fuzzy if-then classification rules from data. Many approaches have been proposed for this purpose including heuristic approaches [3], neuro-fuzzy techniques [4], genetic algorithms [5], etc.

In this paper, a fuzzy rule-based classifier is constructed by finding a compact set of fuzzy if-then rules to perform the classification task. In this way, we generate a set of candidate rules for each class and construct an initial rule-base by selecting a subset of rules. Having an initial rule-base for the problem, we propose an algorithm that attempts to minimize the error-rate of the classifier on training data by adjusting the weight of each fuzzy rule in the rule-base. The algorithm learns the local weight of each fuzzy rule assuming that the weights of all other rules in the rule-base is given and fixed. The resulting weight is locally optimal in the sense that it minimizes the error rate of the classifier in a hill-climbing way. To improve the generalization ability of the classifier, we introduce a weighting mechanism on training examples. We use the class distribution of the neighbors of a training example to determine its weight. Small weight is assigned to an example when the majority of its neighbors are from different classes. The weight assigned to a training example is considered as the cost of misclassification of it and the learning algorithm is modified to minimize the sum of costs for misclassified examples.

The rest of this paper is organized as follows. In section 2, the fuzzy rule-based classifier consisting of fuzzy classification reasoning and rule base construction is described. In section 3, the proposed rule-weight learning algorithm for non-weighted and weighted examples is presented, respectively. In section 4, the experimental results are discussed. Section 5 concludes the paper.

2. Fuzzy Rule-Based Classifier

Assume we have \( m \) training vectors \( X_p = (x_{pi}, \ldots, x_{pm}) \), \( p = 1, 2, \ldots, m \) from \( M \) different classes where \( X_p \) is an \( n \)-dimensional vector of attributes in which \( x_{pi} \) is the \( i \)-th attribute value of the \( p \)-th training vector \( (i = 1, 2, \ldots, n) \). For the classification problem in hand, we use fuzzy if-then rules of the form below:

Rule \( \text{R}_q \): If \( x_{pi} \) is \( A_{q_i} \) and \( \ldots \) and \( x_{pm} \) is \( A_{q_m} \) then class \( C_q \) with \( CF_q \) \hspace{1cm} (1)

where \( \text{R}_q \) is the label of the \( q \)-th fuzzy if-then rule, \( A_{q_i} \) presents an antecedent fuzzy set, \( C_q \) is a class label, and \( CF_q \) is the weight assigned to the \( q \)-th rule. To calculate the compatibility grade [6] of each training vector \( X_p \) with the antecedent part of the rule \( A_{q} = (A_{q_1}, \ldots, A_{q_m}) \), we use the product operator as follows:
\[
\mu_{A_q}(X_p) = \prod_{i=1}^{n} \mu_{A_q}(x_{p_i})
\]  
(2)

where \( \mu_{A_q}(x_{p_i}) \) is the compatibility grade of \( x_{p_i} \) with fuzzy membership function \( A_q \). To determine the consequent class of the \( q \)-th rule \( C_q \), we measure the confidence degree of the association rule \( "A_q \Rightarrow \text{Class } h" \) for each class, where \( A_q \) is a multi dimensional fuzzy set representing the antecedent conditions and \( h \) is a class label. Confidence of a fuzzy association rule \( R_q \) is defined as follows [7]:

\[
c(A_q \Rightarrow h) = \frac{\sum_{x_p \in \text{Class } h} \mu_{A_q}(x_p)}{\sum_{x_p \in \text{Class } h} \mu_{A_q}(x_p), h=1,2,\ldots, M}
\]  
(3)

where \( \mu_{A_q}(x_p) \) is the compatibility grade of vector \( X_p \) with the antecedent part of the rule \( R_q \), \( m \) is the number of training vectors and \( C_q \) is a class label. The class with maximum confidence degree is identified to determine the consequent class \( C_q \):

\[
q = \arg\max_c |c(A_q \Rightarrow \text{Class } h)|, h=1,2,\ldots,M
\]  
(4)

An input vector is classified regarding to the consequent class of the winner rule. By using rules of the form (1), a weight assigned to each rule is used to find the winner rule. Rule weighting has a profound effect on the classification ability of fuzzy classifiers [8]. In this paper, we propose a learning mechanism to find the weight of each rule. The winner rule \( R_w \) is chosen for the input vector \( X_t \) in the following manner:

\[
\mu_{A_q}(X_t) \cdot CF_e = \max\{\mu_{A_q}(X_t) \cdot CF_j | R_j, j=1,2,\ldots,N\}
\]  
(5)

Note that the classification of the vector \( X_t \) not covered by any rule in the rule-base is rejected. The classification of \( X_t \) is also rejected if two rules with different consequent classes have the same value of \( \mu(X_t) \cdot CF \) in (5). To construct the rule-base for the given problem, we follow the study in [7]. Each feature is first normalized into interval \([0, 1]\) and subsequently partitioned using 9 fuzzy sets shown in Figure 1 simultaneously.

![Figure 1. Different partitioning of each feature axis.](image)

Using the training set, an initial rule-base for a problem is constructed by the following steps. First, all rules of length \( \leq 3 \) (i.e. having 3 or less antecedent conditions) are generated as candidate rules. The consequent class of a fuzzy rule is specified by the dominant class in the subspace covered by the rule using (3). In the second step, generated candidate rules are divided into \( M \) groups according to their consequent classes. The candidate rules in each group are sorted in descending order of a rule evaluation metric presented in [9]. With this measure, the evaluation of rule \( R_q \) from class \( C_q \) can be expressed as:

\[
e(R_q) = \sum_{x_p \in \text{Class } C_q} \mu_{A_q}(X_p) - \sum_{x_p \notin \text{Class } C_q} \mu_{A_q}(X_p)
\]  
(6)

In the final step, a rule-base is constructed by selecting the \( Q \) best rules from each class.

3. Rule-weight learning algorithm

We assume that an initial rule-base is constructed using the method described in the previous section. In this section, an iterative learning scheme is proposed that specifies a weight to each fuzzy rule in the rule-base such that the number of misclassified training vectors is minimized. The basic element of this scheme is an algorithm that finds the optimal weight of rule \( R_k \) ∈ rule-base, assuming that the weight of all other rules in the rule-base is given and fixed.

Consider the rule \( R_k \) of the form (1) with class \( T \) as a typical rule in the rule-base. The weight of this rule (given the weight of all other rules) is found such that the error rate on the training data is minimized. For this purpose, in the first step, the rule is removed from the rule-base (i.e. \( CF_k = 0 \)). Then, training examples of class \( T \) that are classified correctly without rule \( R_k \) are removed from the training set. Note that these vectors will be classified correctly regardless of the value of \( CF_k \). Similarly, training examples of class \( \overline{T} \) that are misclassified without rule \( R_k \) are also removed. These vectors will be misclassified regardless of the value of \( CF_k \). For each training vector \( X_t \) left in the training set, the following measure is calculated as its score \( S(X_t) \),

\[
S(X_t) = \frac{\max\{CF_j \cdot \mu_{A_q}(X_t) | Consequent(R_j) \neq \text{Class } T\}}{\mu_{A_q}(X_t)}
\]  
(7)

The algorithm given in Figure 2 is then used to find the weight of rule \( R_k \). This algorithm receives a set of vectors \( X_t \) and their scores \( S(X_t) \) and gives the best threshold (best-th) as the output. The best-th is used as the weight of the rule \( R_k \). This algorithm sorts the vectors in ascending order of their scores. Assuming that a set of \( m \) vectors and their scores is passed to this algorithm, a maximum of \( m+1 \) thresholds are examined to find the best threshold. Note that for each specified threshold \( th \), all training vectors, \( X_t \) with \( S(X_t) < th \) are classified as class \( T \). The best threshold (i.e. weight) is
simply the one that maximizes the classification rate on the listed vectors. The overall procedure given above calculates the optimal weight of a fuzzy rule assuming that the weight of all other rules are given and fixed. The search for the best combination of rule-weights is conducted by optimizing each rule in turn assuming that the order of the rules to be optimized is fixed.

**Inputs:** vectors \( X_i \), scores \( S(X) \)

**Output:** the value of best threshold (best-th)

current = classification accuracy corresponding to the threshold of \( th = 0 \)
(classifying everything as class \( T \))

optimum = current

best-th = 0

rank the vectors in ascending order of their scores

(assume that \( X_i \) and \( X_{i+1} \) are two successive vectors in the list)

for each different threshold \( th = (\text{Score}(X_i) + \text{Score}(X_{i+1}))/2 \)

current = accuracy corresponding to the specified threshold (i.e., all patterns \( X \) having \( \text{Score}(X) < th \) are classified as class \( T \))

if current > optimum then

optimum = current

best-th = th

end if

end for

(assume last is the score of the last vector in the list and \( t \) is a positive number)

current = accuracy corresponding to \( \text{th} = l + t \) (classifying everything as class \( T \))

if current > optimum then

optimum = current

best-th = th

end if

return best-th

**Figure 2.** Algorithm for finding the best threshold

3.1. Weighting training examples

The learning algorithm presented in previous section attempts to tune the fuzzy rule-base by minimizing the number of misclassified training vectors. It implicitly assumes that all training examples are equally important. However, some examples are considered to be noisy as for learning these examples, the learning algorithm would be in contradiction with other training examples or would need to increase its complexity in order to accommodate them. Learning these difficult examples may lead the algorithm to be unable to generalize well. A weighting mechanism can be used to assign a weight in the interval \([0, 1]\) to each training example. Assigning a small weight to a noisy example can reduce its influence.

Assume that for a specific problem, \( m \) training examples are given. We use the class distribution in \( k \) nearest neighbor of a training example to specify its weight. The distance metric that we use for this purpose is Euclidean distance function. The weight of training example \( X \) is found by the following measure:

\[
w_x = \min \left\{ \frac{k_{\text{same}}}{k}, k = 1, 2, \ldots, k_{\text{max}} \right\}
\]

where \( k \) is the number of neighbors, \( k_{\text{same}} \) is the number of neighbors having the same class as \( X \), and \( k_{\text{max}} \) identifies the limit for the neighborhood size. With this measure, a small weight is assigned to an example if the majority of its neighbors are from a different class. We use \( k_{\text{max}} = 5 \) in all experiments in the next section.

The mechanism for weighting training examples is regarded as a preprocessing step prior to classification. The weight assigned to a training example is considered as the cost for misclassification of it and the algorithm of previous section is modified to minimize the sum of cost for misclassified examples. For a set of \( m \) training examples \( \{X_p, p = 1, \ldots, m\} \), the cost function is formally specified as:

\[
Cost = \sum_{p=1}^{m} w_p \cdot C(X_p)
\]

where \( w_p \) is the weight assigned to training example \( X_p \) and \( C(X_p) \) is defined as:

\[
C(X_p) = \begin{cases} 
1 & \text{if } X_p \text{ is misclassified} \\
0 & \text{otherwise}
\end{cases}
\]

4. Experimental results

In order to evaluate the performance of the proposed method, two sets of experiments are conducted. The designed classifier is validated on six real world datasets chosen from UCI Machine Learning Database Repository [10]. Table 1 indicates the input and output characteristics of the datasets as well as the dataset size. We also examine the framewise phone classification accuracy of the proposed fuzzy classifier on TIMIT speech corpus [11].

<table>
<thead>
<tr>
<th>Data set</th>
<th># attributes</th>
<th># patterns</th>
<th># Classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cancer</td>
<td>9</td>
<td>699</td>
<td>2</td>
</tr>
<tr>
<td>Iris</td>
<td>4</td>
<td>150</td>
<td>3</td>
</tr>
<tr>
<td>Glass</td>
<td>10</td>
<td>214</td>
<td>6</td>
</tr>
<tr>
<td>Sonar</td>
<td>60</td>
<td>208</td>
<td>2</td>
</tr>
<tr>
<td>Wine</td>
<td>13</td>
<td>178</td>
<td>3</td>
</tr>
<tr>
<td>Pima</td>
<td>8</td>
<td>768</td>
<td>2</td>
</tr>
</tbody>
</table>

Our objective is to investigate the effect of the proposed learning algorithm on the generalization ability of fuzzy rule-based classifier. We first scale each attribute to the interval \([0, 1]\). Using the method of section 2, we generate all fuzzy rules of having 1, 2 and 3 antecedent conditions and construct an initial rule-base by selecting \( Q = 200 \) best rules from each class using (6) as the rule evaluation metric. Table 2 reports classification results of the fuzzy classifier with no weighting mechanism attached, rule-weight learning mechanism of section 3, and rule-weight learning on weighted examples, respectively. We used the measure (8) to determine the weight of each training example and subsequently used (9) as the cost function to be
minimized by the learning algorithm. All the reported results are the average of ten trials of ten-fold cross validation.

Table 2. Results achieved by the fuzzy rule-based classifier

<table>
<thead>
<tr>
<th>Fuzzy classifier with</th>
<th>Data set</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cancer</td>
</tr>
<tr>
<td>no weighting</td>
<td>7.24</td>
</tr>
<tr>
<td>mechanism</td>
<td></td>
</tr>
<tr>
<td>weighted rules</td>
<td>4.14</td>
</tr>
<tr>
<td>weighted rules+</td>
<td>3.08</td>
</tr>
<tr>
<td>weighted examples</td>
<td></td>
</tr>
</tbody>
</table>

We can observe considerable improvement on the generalization ability of the classifier by applying the rule-weight learning mechanism. It can also be seen that weighting training examples has further improved the performance of the classifier.

4.1. Phone classification on TIMIT corpus

We conducted the phone classification experiment using the TIMIT corpus [11] because of its high-quality phone labels. All the results reported are framewise classification accuracies for the complete test set (the 1344 si and sx sentences). The speech waveforms are parameterized by a standard Mel-Frequency Cepstral Coefficient (MFCC) front end. The cepstral analysis uses a 25 msec Hamming window with a frame shift of 10 msec. Each input pattern $X_i$ consists of the current frame of 12 MFCCs as well as the zeroth cepstral coefficient with the first and second time derivatives of the cepstra, and two context frames on each side, making a total of $(13+13+13) * 3 = 195$ components. This formulation was arrived at by experimentation with varying numbers of context frames left and right of the frame being classified. The training set has about 1.1 million frames and the test set has about 400 thousand frames. Each frame has an associated 1-of-48 phonetic label derived from the TIMIT label files.

In Table 4, the proposed method is compared to the previous works on the TIMIT phone classification.

Table 3. Framewise phone classification on TIMIT data set

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Error Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recurrent Neural Nets (Schuster) [12]</td>
<td>34.7</td>
</tr>
<tr>
<td>Bidirectional LSTM (Graves) [13]</td>
<td>29.8</td>
</tr>
<tr>
<td>Fuzzy IRL (Dezhangi) [14]</td>
<td>29.4</td>
</tr>
<tr>
<td>SVM (Salomon) [15]</td>
<td>28.6</td>
</tr>
<tr>
<td>This paper</td>
<td>27.8</td>
</tr>
</tbody>
</table>

Table 4 shows that the proposed fuzzy classifier outperformed previous works we found in the literature on this task while preserving interpretability of the classifier output such that a human user can understand and infer from the resulting rule-base.

5. Conclusion

In this paper, we proposed a method of learning rule-weights in fuzzy rule-based classifiers. The learning mechanism attempts to minimize the number of misclassified patterns in the training data. To further improve the generalization ability, a weighting scheme was applied on training examples. The weight assigned to each training example was considered as the cost of misclassification of it. The proposed learning algorithm was then modified to minimize the sum of cost for misclassified patterns. The experimental results using six data sets from UCI ML repository along with TIMIT speech corpus illustrated that our proposed cost-minimizing algorithm is effective to improve the performance of the fuzzy rule-based classifier.

6. References