Learning Fuzzy If-Then Rules for Pattern Classification with Weighted Training Patterns

Tomoharu Nakashima, Yasuyuki Yokota, Hisao Ishibuchi
College of Engineering, Osaka Prefecture University
Gakuencho 1-1, Sakai, Osaka 599-8531, Japan
{nakashi@, yokota@ci, hisaoi@}.cs.osakafu-u.ac.jp

Gerald Schaefer
School of Computing and Informatics, Nottingham Trent University
Burton Street, Nottingham, NG1 4BU, UK
Gerald.Schaefer@ntu.ac.uk

Abstract

In this paper we propose a learning method of fuzzy if-then rules for pattern classification problems. We assume that each training pattern has a weight that describes its importance. The antecedent part of fuzzy if-then rules are specified by partitioning each attributes into fuzzy sets while the consequent class and the degree of certainty of the fuzzy if-then rules are determined from the compatibility and weights of training patterns. The proposed learning method adjusts the degree of certainty so that the classification cost is minimized. Experimental results on several UCI data sets show the effectiveness of the proposed method.

Keywords: Fuzzy if-then rule, pattern classification, learning, error correction.

1 Introduction

While fuzzy rule-based systems have been mainly applied to control problems [10, 11, 14] in the past, recently they have also been applied to pattern classification problems. Various methods have been proposed for the automatic generation of fuzzy if-then rules from numerical data for pattern classification [13, 9, 7, 1, 2, 3].

Let us assume that our pattern classification problem is an n-dimensional problem with M classes and m given training patterns $\mathbf{x}_p = (x_{p1}, x_{p2}, \ldots, x_{pn})$, $p = 1, 2, \ldots, m$. Without loss of generality, we assume each attribute of the given training patterns to be normalised into the unit interval $[0, 1]$; that is, the pattern space is an n-dimensional unit hypercube $[0, 1]^n$. In this study we use fuzzy if-then rules of the following type as a base of our fuzzy rule-based classification systems:

$$\text{Rule } R_j\text{: If } x_1 \text{ is } A_{j1} \text{ and } \ldots \text{ and } x_n \text{ is } A_{jn} \text{ then Class } C_j \text{ with } CF_j, \quad j = 1, 2, \ldots, N, \tag{1}$$

where R_j is the label of the j-th fuzzy if-then rule, $A_{j1}, \ldots, A_{jn}$ are antecedent fuzzy sets on the unit interval $[0, 1]$, C_j is the consequent class (i.e. one of the M given classes), and CF_j is the grade of certainty of the fuzzy if-then rule R_j. As antecedent fuzzy sets we use triangular fuzzy sets as in Figure 1 where we show a partition of the unit interval into a number of fuzzy sets.

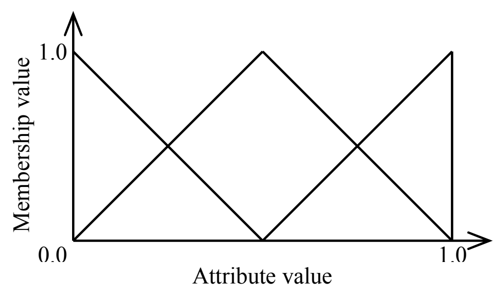

Figure 1: Membership function.

There are several cases where misclassification/rejection of a particular input pattern will cause extra costs. For example in medical diagnosis of cancer, diagnosing people with cancer as no having the disease chould be penalised more than diagnosing healthy individuals as cancer candidates. In [12] a pattern classification problem is re-formulated as a cost minimization problem. The concept of a weight is introduced for each training pattern in order to handle this situation.

The weight of an input pattern can be viewed as the cost of misclassification/rejection for it. Fuzzy if-then rules are generated by considering the weights as well as the compatibility of training patterns.

In this paper we propose a learning method for fuzzy if-then rules. We adjust the grades of certainty CF_j in Equation (1). The proposed method can be categorized as error-correction type learning. That is, the adjustment of fuzzy if-then rules occurs only when a training pattern is misclassified. The main idea is to penalise fuzzy if-then rules that misclassify a training pattern and to enhance fuzzy if-then rules that are used to correctly classify the pattern.

In a series of computer simulations we examine the performance of the proposed learning method. The classification ability is exmined for several real-world pattern classification that are available from the UCI machine learning repository. From the simulation results we show that the performance of a fuzzy classification system is improved.

2 Fuzzy Classification

This section describes the generation of fuzzy if-then rules from given training patterns. Our fuzzy rule-based classification system consists of N fuzzy if-then rules each of wich has a form as in Equation (1). There are two steps in the generation of fuzzy if-then rules: specification of antecedent part and determination of consequent class C_j and the grade of certainty CF_j. The antecedent part of fuzzy if-then rules is specified manually. Then the consequent part (i.e. consequent class and the grade of certainty) is determined from the given training patterns [7]. In [6] it is shown that the use of the grade of certainty in fuzzy if-then rules allows us to generate comprehensible fuzzy rule-based classification systems with high classification performance.

2.1 Fuzzy Rule Generation

Let us assume that m training patterns $\mathbf{x}_p = (x_{p1}, \ldots, x_{pn})$, $p = 1, \ldots, m$, are given for an n-dimensional C-class pattern classification problem. We also assume that a weight ω_p, $p = 1, \ldots, m$, is assigned to each training pattern a priori. The consequent class C_j and the grade of certainty CF_j of the if-then rule are determined in the following manner:

Step 1: Calculate $\beta_{\text{Class } h}(j)$ for Class h as

$$\beta_{\text{Class } h}(j) = \sum_{\mathbf{x}_p \in \text{Class } h} \mu_j(\mathbf{x}_p) \cdot \omega_p, \quad (2)$$

where

$$\mu_j(\mathbf{x}_p) = \mu_{j1}(x_{p1}) \cdot \ldots \cdot \mu_{jn}(x_{pn}), \quad (3)$$

and $\mu_{jn}(\cdot)$ is the membership function of the fuzzy set A_{jn}. In this paper, we use triangular fuzzy sets as in Figure 1.

Step 2: Find Class $\hat{h}$ that has the maximum value of $\beta_{\text{Class } h}(j)$:

$$\beta_{\text{Class } \hat{h}}(j) = \max_{1 \le k \le C} \{\beta_{\text{Class } k}(j)\}. \quad (4)$$

We note that this fuzzy rule generation method can also be applied to the standard pattern classification problem where there are no pattern weights. In this case, the class and the grade of certainty are determined from training patterns by specifying a pattern weight as $\omega_p = 1$ for $p = 1, \ldots, m$.

If two or more classes take the maximum value, the consequent class C_j of the rule R_j can not be determined uniquely. In this case, specify C_j as $C_j = \phi$. If a single class $\hat{h}$ takes the maximum value, let C_j be Class $\hat{h}$. The grade of certainty CF_j is determined as

$$CF_j = \frac{\beta_{\text{Class } \hat{h}}(j) - \bar{\beta}}{\sum_h \beta_{\text{Class } h}(j)} \quad (5)$$

with

$$\bar{\beta} = \frac{\sum_{h \neq \hat{h}} \beta_{\text{Class } h}(j)}{c - 1}. \quad (6)$$

2.2 Fuzzy Reasoning

Using the rule generation procedure outlined above we can generate N fuzzy if-then rules as in Equation (1). After both the consequent class C_j and the grade of certainty CF_j are determined for all N rules, a new pattern $\mathbf{x} = (x_1, \ldots, x_n)$ can be classified by the following procedure:

Step 1: Calculate $\alpha_{\text{Class } h}(\mathbf{x})$ for Class h, $j = 1, \ldots, C$, as

$$\alpha_{\text{Class } h}(\mathbf{x}) = \max\{\mu_j(\mathbf{x}) \cdot CF_j | C_j = h\}, \tag{7}$$

Step 2: Find Class h' that has the maximum value of $\alpha_{\text{Class } h}(\mathbf{x})$:

$$\alpha_{\text{Class } h'}(\mathbf{x}) = \max_{1 \leq k \leq C}\{\alpha_{\text{Class } k}(\mathbf{x})\}. \tag{8}$$

If two or more classes take the maximum value, then the classification of $\mathbf{x}$ is rejected (i.e. $\mathbf{x}$ is left as an unclassifiable pattern), otherwise assign $\mathbf{x}$ to Class h'.

3 Learning Fuzzy If-Then Rules

This section presents a learning method of fuzzy if-then rules for improving classification performance. It adjusts the grades of certainty CF_j. We do not adjust the shape of membership function as the interpretability of fuzzy if-then rules would be reduced by doing so. The proposed learning method is based on an error-correction learning approach where the adjustment occurs when classification of training patterns is not successful. When a training pattern is correctly classified we do not adjust the grade of certainty. The main idea of the learning method is to adjust the degree of certainty CF_j of two fuzzy if-then rules: We decrease the degree of certainty of a fuzzy if-then rule that misclassifies a training pattern and in turn increase that of a fuzzy if-then rule that is supposed to correctly classify the training pattern.

Let us assume that we have generated fuzzy if-then rules by the rule-generation procedure detailed in Section 2.1. We also assume that a fuzzy if-then rule R_j misclassifies a training pattern $\mathbf{x}_p$. That is, R_j is used to classify $\mathbf{x}_p$ from Class c^* by using Equation (8) but the consequent class C_j does not agree with the true class of the training pattern $\mathbf{x}$. Let R_* be the fuzzy if-then rule that is selected by Equation (7). That is, R_* has the maximum value of $\alpha_{\text{Class } c^*}(\mathbf{x}_p)$ among those fuzzy if-then rules with Class c^* but does not have the maximum value among all generated fuzzy if-then rules. The proposed learning method adjusts the grades of certainty of R_j and R_* as follows:

$$CF_j^{new} = CF_j^{old} - \eta \cdot \omega_p \cdot CF_j^{old}, \tag{9}$$

$$CF_*^{new} = CF_*^{old} - \eta \cdot \omega_p \cdot (1 - CF_*^{old}), \tag{10}$$

where ω_p is the weight of the training pattern $\mathbf{x}_p$, and η is a positive constant value. We assume that $0 \leq \eta \leq 1$.

One epoch of the proposed learning method involves examining all given training patterns. Thus there will be $2m$ adjustments of fuzzy if-then rules if all m training patterns are misclassified. The learning process is summarized as follows:

Step 1: Generate fuzzy if-then rules from m given training patterns by the procedure in Section 2.1.

Step 2: Set K as $K = 1$.

Step 3: Set p as $p = 1$.

Step 4: Classify $\mathbf{x}_p$ by using the generated fuzzy if-then rules in Step 1.

Step 5: If $\mathbf{x}_p$ is misclassified, adjust the grades of certainty using Equations (9) and (10). Otherwise no rules are adjusted.

Step 6: If $p < m$, let $p := p + 1$ and go to Step 4. Otherwise go to Step 7.

Step 7: If K reaches a pre-specified value, stop the learning procedure. Otherwise let $K := K + 1$ and go to Step 3.

Note that K in the above learning procedure corresponds to the number of epochs.

4 Experimental results

4.1 Cost function

In this section we examine the performance of the proposed method. Under the assumption that a weight is assigned to each training pattern, we use the concept of classification/rejection cost throughout the computer simulations in this paper. The weight of training patterns can be viewed as the importance of the patterns. More emphasis should be placed on those patterns with large weights than on those with small weights. The weight of misclassified/rejected patterns is considered as a cost of misclassification or rejection. We define a cost function $\texttt{Cost}(S)$ of a fuzzy classificatoin system S as follows:

$$\texttt{Cost}(S) = \sum_{p=1}^{m} \omega_p \cdot z_p(S), \tag{11}$$

where m is the number of training patterns, ω_p is the weight of the training pattern $\mathbf{x}_p$, and $z_p(S)$ is a binary variable set according to the classification result of the training pattern $\mathbf{x}_p$ by S: $z_p(S) = 0$ if $\mathbf{x}_p$ is correctly classified by S, and $z_p(S) = 1$ otherwise (i.e. $\mathbf{x}_p$ is misclassified or rejected). We use this cost function as a performance measure as well as classification rate.

4.2 Assigning weights

We use real-world pattern classification problems that are commonly used in literature. All the classification problems are available from the UCI machine learning repository. Since weights for training patterns are not included in these pattern classification sets, we assign a weight to each training pattern in order to make a synthetic situation where a weight is given a priori. We consider two cases. In Case 1 we specify ω_p as $\omega_p = 0.5$ if the training pattern $\mathbf{x}_p$ belongs to Class 1 and $\omega_p = 1.0$ if $\mathbf{x}_p$ belongs to Class 2. On the other hand ω_p of the training pattern $\mathbf{x}_p$ is specified as $\omega_p = 1.0$ if $\mathbf{x}_p$ belongs to Class 1 and $\omega_p = 0.5$ if $\mathbf{x}_p$ belongs to Class 2. Note that we apply the proposed learning method only to 2-class classification problems in the computer simulations in this paper. However, it is clear that it can similarly be applied to pattern classification problems with any number of classes.

Table 1: Data sets used in the experiments.

Data set	Attribute	Classs	Pattern
Haberman	3	2	306
Breast cancer	9	2	683
Sonar	60	2	208

4.3 Experimental settings

In order to construct a fuzzy classification system it should be determined how to partition each attribute variable into fuzzy sets. In this paper we divide each axis into three fuzzy sets as shown in Figure 1. The number of generated fuzzy if-then rules in a fuzzy classification system depends on the partition of attributes and the dimensionality of a pattern classsifiation problem. Since there are three fuzzy sets for each attribute, the possible number of combination of antecedent fuzzy sets is $N = 3^n$ where n is the number of attributes.

Table 2: Simulation results before learning.

Data	Classification rate	Cost(S)
Haberman	74.2%	40.5
Breast cancer	98.2%	9.0
Sonar	80.3%	38.0

Table 3: Simulation results ($\eta = 0.1, K = 10$).

Data	Classification rate	Cost(S)
Haberman	74.2%	39.5
Breast cancer	98.7%	5.5
Sonar	89.9%	12.0

Table 4: Simulation results ($\eta = 0.1, K = 20$).

Data	Classification rate	Cost(S)
Haberman	73.9%	40.5
Breast cancer	98.8%	5.0
Sonar	92.3%	9.5

There are two parameters in the proposed learning method: η, a positive constant and the number of epochs K. For the computer simulations reported here we used the following parameter specifications: $\eta = 0.1, 0.3$, and $K = 10, 20$, i.e. four possible combinations.

We used three real-world data sets from the UCI machine learning repository. Table 1 lists the details of these data sets.

4.4 Results for Case 1

The performance of fuzzy classification systems before applying the learning method is shown in

Table 5: Simulation results ($\eta = 0.3, K = 10$).

Data	Classification rate	Cost(S)
Haberman	73.9%	40.0
Breast cancer	98.8%	5.0
Sonar	89.9%	12.0

Table 6: Simulation results ($\eta = 0.3, K = 20$).

Data	Classification rate	Cost(S)
Haberman	73.9%	40.0
Breast cancer	98.8%	5.0
Sonar	92.8%	9.0

Table 2. We also show the simulation results for Case 1 ($w_q = 0.5$ for Class 1 patterns and $w_q = 1.0$ for Class 2) in Tables 3 to 6. From there we can see that the performance of fuzzy classification systems was clearly improved by the learning fuzzy if-then rules. We can also see that the effect of the learning method is larger when larger values of η and K are used.

4.5 Results for Case 2

As mentioned Case 2 uses $w_q = 1.0$ for Class 1 patterns and $w_q = 0.5$ for Class 2 patterns. In Table 7 the performance of fuzzy classification systems before learning is shown. Note that classification/rejection costs ($\texttt{Cost}(S)$) are different from those in Table 2 although classification rates are the same. This is because the number of Class 1 patterns is different from that of Class 2 patterns. The performance of the proposed learning method is shown in Tables 8 to 11. Again it is apparent that the performance of fuzzy classification systems is improved. In fact, the effect of the learning method for Case 2 is more significant than for Case 1.

Table 7: Simulation results before learning.

Data	Classification rate	Cost(S)
Haberman	74.2%	78.0
Breast cancer	98.2%	9.0
Sonar	80.3%	23.5

Table 8: Simulation results ($\eta = 0.1, K = 10$).

Data	Classification rate	Cost(S)
Haberman	77.1%	59.0
Breast cancer	99.0%	4.0
Sonar	74.0%	27.0

Table 9: Simulation results ($\eta = 0.1, K = 20$).

Data	Classification rate	Cost(S)
Haberman	76.8%	60.0
Breast cancer	99.0%	4.0
Sonar	85.1%	15.5

5 Conclusions

In this paper, we have proposed a learning method for fuzzy rules that adjusts the grade of certainty. The main idea of the proposed method is to penalise a fuzzy if-then rule that is responsible for the misclassification of a training pattern and to enhance a fuzzy if-then rule that is supposed to correctly classify the training pattern. Experimental results based on various standard data sets have demonstrated the usefulness of this approach.

Currently we are examining the performance of the proposed method on unseen test data. Although this paper examined classification performance on training patterns the final aim of classification is to correctly classify unseen patterns.

References

[1] M. Grabisch. 1996, “The representation of importance and interaction of features by fuzzy measures”, Pattern Recognition Letters, Vol. 17:567-575.

[2] M. Grabisch, F. Dispot. 1992, “A comparison of some methods of fuzzy classification on real data”, Proc. of 2nd Intl. Conf. on Fuzzy Logic and Neural Networks, 659-662.

[3] M. Grabisch, and J.-M. Nicolas. 1994, “Classification by fuzzy integral: performance and tests”, Fuzzy Sets and Systems, Vol. 65, No. 2/3:255-271.

Table 10: Simulation results ($\eta = 0.3, K = 10$).

Data	Classification rate	Cost(S)
Haberman	76.5%	63.5
Breast cancer	99.3%	3.0
Sonar	86.1%	14.5

Table 11: Simulation results ($\eta = 0.3, K = 20$).

Data	Classification rate	Cost(S)
Haberman	77.12%	61.5
Breast cancer	99.6%	1.5
Sonar	88.0%	12.5

[4] H. Ishibuchi, T. Nakashima. 1999a, "Performance evaluation of fuzzy classifier systems for multi-dimensional pattern classification problems", IEEE Trans. on Syst., Man, Cybernetics, Part B Vol. 29:601-618.

[5] H. Ishibuchi, T. Nakashima. 1999b, "Improving the performance of fuzzy classifier systems for pattern classification problems with continuous attributes", IEEE Trans. on Industrial Electronics, Vol. 46, No. 6:1057-1068.

[6] H. Ishibuchi, T. Nakashima. 2001, "Effect of rule weights in fuzzy rule-based classification systems", IEEE Trans. on Fuzzy Systems, Vol. 9, No. 4:506-515.

[7] H. Ishibuchi, K. Nozaki, H. Tanaka. 1992, "Distributed representation of fuzzy rules and its application to pattern classification", Fuzzy Sets and Systems, Vol. 52, No. 1:21-32.

[8] H. Ishibuchi, K. Nozaki, N. Yamamoto, H. Tanaka. 1995, "Selecting fuzzy if-then rules for classification problems using genetic algorithms", IEEE Trans. on Fuzzy Systems, Vol. 3, No. 3:260-270.

[9] G.J. Klir, B. Yuan. 1995, Fuzzy Sets and Fuzzy Logic, Prentice-Hall.

[10] C.C. Lee. 1990, "Fuzzy Logic in Control Systems: Fuzzy Logic Controller Part I and Part II", IEEE Trans. Syst., Man, Cyberetics, Vol. 20:404-435.

[11] C.T. Leondes. 1999, "Fuzzy theory Systems", Techniques and Applications. Academic Press, San Diego, Vol.1-4, 1999.

[12] T. Nakashima, Y. Yokota, H. Ishibuchi, A. Bargiela. 2005, "Constructing Fuzzy Classification Systems from Weighted Training Patterns," Proc. of European Simulation Conference, in press.

[13] K. Nozaki, H. Ishibuchi, H. Tanaka. 1996, "Adaptive fuzzy rule-based classification systems", IEEE Trans. on Fuzzy Systems, Vol. 4, No. 3:238-250.

[14] M. Sugeno. 1985, "An Introductory Survey of Fuzzy Control", Information Science, Vol. 30, No. 1/2:59-83.