

A Tolerant Rough Set Approach for Handwritten Numeral Character Classification

Daijin Kim†, Chul-Hyun Kim†
† Dept. of Computer Engineering, DongA University
Pusan, 604-714, Korea

Abstract— This paper¹ proposes a new data classification method based on the tolerant rough set that extends the existing equivalent rough set. Similarity measure between two data is described by a distance function of all constituent attributes and they are defined to be tolerant when their similarity measure exceeds a similarity threshold value. The determination of optimal similarity threshold value is very important for the accurate classification. So, we determine it optimally by using the genetic algorithm (GA), where the goal of evolution is to balance two requirements such that (1) some tolerant objects are required to be included in the same class as many as possible and (2) some objects in the same class are required to be tolerable as much as possible. After finding the optimal similarity threshold value, a tolerant set of each object is obtained and the data set is grouped into the lower and upper approximation set depending on the coincidence of their classes. We propose a two-stage classification method that all data are classified by using the lower approximation at the first stage and then the non-classified data at the first stage are classified again by using the rough membership functions obtained from the upper approximation set. We apply the proposed classification method to the handwritten numeral character classification problem and compare its classification performance and learning time with those of the feedforward neural network's back-propagation algorithm.

1 Introduction

Pattern classification is a problem that partitions a data space \mathfrak{R}^n into classes and then assigns a point $x \in \mathfrak{R}^n$ to one of those classes. Many application examples have been found in engineering applications such as the OCR recognition, fingerprint and facial recognition, in industrial applications such as part classification in computer vision, and in medical applications such as the blood and electrocardiogram analysis, and so on. Pattern classification methods are classified into three different ways : statistical pattern classification, syntactic pattern classification, and neural network-based pattern recognition.

Recent researches of pattern classification are based on the neural-networks as follows.

¹This work was supported by a grant of IITA of Republic of Korea(1997).

Carpenter and Grossberg [1] developed a fast and reliable analog pattern clustering system called fuzzy ART (Adaptive Resonance Theory) that combined fuzzy logic with the ART1. Lin and Lee [2] introduced a general neural-network model for fuzzy logic control and decision systems that could train fuzzy logic rules and optimal input/output membership functions. P. Simpson [3] developed a fuzzy min-max classification neural network that utilized fuzzy sets as pattern classes, where learning in the neural network was performed by properly placing and adjusting hyperboxes in the pattern space. Since the above classifiers have a connectionist structure that combines fuzzy logic and neural network, they are destined to have the same drawbacks as neural networks such that (1) the possibility of having the nonconvergent solution due to the wrong choice of initial weights, (2) relatively long learning time, and (3) the possibility of having the unoptimal solution due to the local minima problem. To meet the above requirements for an ideal classifier as much as possible and overcome some drawbacks of the above neuro-fuzzy classifiers, we propose a new classification method based on the tolerant rough set.

This paper is organized as follows. Section II explains a theoretical background of rough set theory, a tolerant rough set appropriate for pattern classification, and a similarity measure used for pattern classification. Section III describes how to determine the similarity threshold value optimally by using the GA. Section IV presents a proposed two-stage classification method based on the lower and upper approximation set. Section V presents the simulation results of applying the proposed classification method to the handwritten numeral character recognition problem and compares its classification performance and learning speed with those of the multilayer feedforward neural networks. Finally, a conclusion is drawn.

2 Tolerant Rough Set

Rough set theory introduced by Z. Pawlak in the early 1980s [4], is a new mathematical tool to deal with vagueness and uncertainty in the areas of machine learning, knowledge acquisition, decision analysis, knowledge discovery from databases, expert systems, decision support systems, inductive reasoning, and pattern recognition. The rough set is based on the indiscernibility relation that satisfies reflexivity, symmetry, and transitivity. However, in the problem of data classification, it is inconvenient to describe the similarity among data with the indiscernibility relation because two data x and z can not be guaranteed in the same class even though a couple of data x and

y are contained in the same class and another couple of data y and z are also contained in the same class. In other word, the transitivity property is not always useful in the problem of classification. This nontransitivity property is more salient for the data within the boundary region. So, we introduce a tolerant relation that is appropriate for the data classification problem because it satisfies only the reflexive and symmetric property [5].

Let $\mathbf{A} = (U, A \cup d)$ be a decision table [6]. Here, U is a set of elements (objects, examples), A is a set of condition attributes, where each attribute $a \in A$ has a set of attribute values V_a , and the set $\{d\}$ is a decision set such as $d = \{1, 2, \dots, r(d)\}$, where $r(d)$ be a number of decision classes. Let $\mathfrak{R}_A = \{R_a : R_a \subseteq V_a \times V_a \wedge a \in A\}$ be a set of tolerant relations. Then each such tolerance relation satisfies

$$\begin{aligned} \text{reflexive} & : \forall v_1 \in V_a, v_1 R_a v_1 \\ \text{symmetric} & : v_1 R_a v_2 \rightarrow v_2 R_a v_1, \end{aligned} \quad (1)$$

where v_1 and v_2 are some attribute values in V_a . We say that two objects x and y are similar to with respect to the attribute a when the attribute values $a(x)$ and $a(y)$ satisfy $a(x) R_a a(y)$. Further, we say that two objects x and y are similar with respect to all attributes A , when they satisfy the tolerance relation with respect to all attributes, i.e. $\forall a \in A, a(x) R_a a(y)$. Hereafter, we denote the above similarity between two objects x and y with respect to all attributes A as $x \tau_A y$ in order to emphasize the tolerance relation.

A tolerance set $TS(x)$ of an object x is defined by a set of all objects that has the tolerance relation with the object x with respect to all attributes as

$$TS(x) = \{y \in U \mid x \tau_A y\}. \quad (2)$$

Then we can define the lower approximation $\underline{\tau}_A(Y)$ and the upper approximation $\overline{\tau}_A(Y)$ of a set $Y \subseteq U$ that have the tolerance relation with respect to all attributes A as

$$\begin{aligned} \underline{\tau}_A(Y) & = \bigcup_{x \in U} \{TS(x) \mid TS(x) \subseteq Y\} \\ \overline{\tau}_A(Y) & = \bigcup_{x \in U} \{TS(x) \mid TS(x) \cap Y \neq \phi\} \end{aligned} \quad (3)$$

The meaning of two approximations in the tolerant relation is similar to that in the indiscernibility relation.

To construct a tolerance relation among the data, we need to define a similarity measure that quantities the closeness between attribute values of objects. Let the similarity measure with respect to the attribute a between two objects x and y be $S_a(x, y)$. Then, two objects are similar with respect to the attribute a when $S_a(x, y) \geq t(a)$, where $t(a)$ is a similarity threshold value of the attribute a whose value is in the interval of $t(a) \in [0, 1]$. So, we can relate the tolerance relation with the similarity measure as

$$a(x) R_a a(y) \iff S_a(x, y) \geq t(a). \quad (4)$$

In the pattern classification problem, the commonly used similarity measure is based on

a normalized distance function as

$$S_a(x, y) = 1 - \frac{d(a(x), a(y))}{d_{max}}, \quad (5)$$

where d_{max} is the maximum value of distance between two attribute values $a(x)$ and $a(y)$. The choice of distance function depends on the type of application. In this work, we select the absolute difference between attribute values as $d(a(x), a(y)) = |a(x) - a(y)|$ due to its computational simplicity.

Next, we can extend the similarity measure $S_A(x, y)$ between two objects x and y with respect to all attributes by an arithmetic average of similarity measures of all attributes as

$$S_A(x, y) = \frac{1}{|A|} \sum_{\forall a \in A} S_a(x, y), \quad (6)$$

where $|A|$ is the number of attributes in A . In the case of considering all attributes A at the same time, we can relate the tolerance relation with the similarity measure as

$$x \tau_A y \iff S_A(x, y) \geq t(A), \quad (7)$$

where $t(A) \in [0, 1]$ is a similarity threshold for pattern classification based on the all attributes A . One of the most important tasks in the pattern classification using the similarity measure defined above is the optimal determination of the similarity threshold $t(A)$, because its proper determination affects the classification performance greatly. In this work, we apply the GA to solve this optimization problem.

3 Determination of Similarity Thresholds

GAs [7] are any population-based iterative adaptive algorithms that use selection, recombination, and mutation operations based on natural selection and biological genetics. GAs have been proven to be powerful methods in search, optimization and machine learning [8]. They encode a potential solution to a specific problem on a simple chromosome-like data structure and apply recombination operators to these structures to achieve optimization.

3.1 Chromosome Representation

When we are applying the GA to determine the optimal similarity threshold values, the inputs into the GA are the information table $\mathbf{A} = (U, A \cup d)$ and the similarity measure $S_a : V_a \times V_a \rightarrow [0, 1]$, and the output from the GA is a set of optimal similarity threshold values $\{t(A) \cup \{t(a) : a \in A\}\}$. So, when an object is represented by n attributes, the chromosome for the GA consists of $n + 1$ consecutive real numbers of the similarity thresholds $\{t(a_1), t(a_2), \dots, t(a_n), t(A)\}$, where $t(a_i)$, ($i = 1, 2, \dots, n$) represents the similarity threshold for the i th attribute, and the last value $t(A)$ represents the similarity threshold that defines the tolerance relation when all attributes A are considered together. We adopt a real number representation of chromosomes since each gene value in the chromosome is a real number.

3.2 Initial Population Generation

The initial gene values in the chromosome are obtained by generating $n + 1$ real-valued random numbers in the interval of $[0.5, 1.0]$. The reason for choosing the interval $[0.5, 1.0]$ as the initial similarity threshold values is that two objects are believed similar when the similarity threshold value between two objects is greater than at least 0.5. We complete initial population by repeating the above operation $|P|$ times, where $|P|$ is the population size.

3.3 Fitness Function

Before considering the fitness function for the optimal determination of the similarity thresholds, we consider a notion of connections to express the indiscernibility of objects. The notion of connections is based on a very simple observation that if $x \in TS(y) \wedge y \in TS(x)$, then we can say that there is a connection between two objects x and y . From this, we define two kinds of connections between two objects x and y as

Good conn. $\leftrightarrow x \in TS(y) \wedge d(x) = d(y)$,

Bad conn. $\leftrightarrow x \in TS(y) \wedge d(x) \neq d(y)$

where $d(x)$ and $d(y)$ are the class decisions of two objects x and y , respectively. When two objects are tolerant and contained in the same (or different) class, they have good (or bad) connection. Fig. 1 illustrates two kinds of connections between objects.

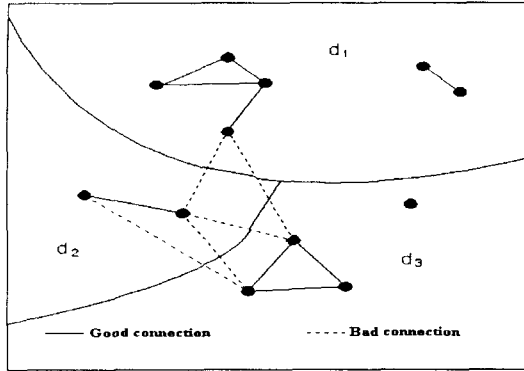


Figure 1: Good connections and bad connections.

When we choose the fitness function for the optimal determination of the similarity thresholds, we consider the following two requirements. First, it is required that some objects that are tolerant each other are included in the same class as many as possible. To meet this requirement, we define a quality of approximation of classification $\gamma_{\tau_A, \{d\}}$ that expresses the ratio of all τ_A - correctly classified objects to all objects. Let a set of objects contained in the same class be $Y_i = \{x \in U | d(x) = d_i, i = 1, 2, \dots, r(d)\}$, where $r(d)$ is the number of decision classes. Consider the tolerance set $TS(x)$ of an object x whose all elements in $TS(x)$ is contained in the same class d_i , i.e. $\{TS(x) | \exists i TS(x) \subseteq Y_i\}$. Then, the union

of such tolerant set with respect to all object U is called a τ_A - positive region of partition $\{Y_i, i = 1, 2, \dots, r(d)\}$, which is defined as

$$POS_{\tau_A, \{d\}} = \bigcup_{x \in U} \{TS(x) | \exists i TS(x) \subseteq Y_i\}. \quad (9)$$

Then, the quality of approximation of classification $\gamma_{\tau_A, \{d\}}$ is defined by the ratio of all τ_A - correctly classified objects to all objects as

$$\gamma_{\tau_A, \{d\}} = \frac{card(POS_{\tau_A, \{d\}})}{card(U)}. \quad (10)$$

As the similarity threshold values increases, the quality of approximation of classification $\gamma_{\tau_A, \{d\}}$ increases because the size of tolerant sets decreases with the increase of the value of similarity thresholds, and thus the chance of inclusion of tolerance sets into the partition set $\{Y_i\}$ becomes larger.

When we consider only the quality of approximation of classification $\gamma_{\tau_A, \{d\}}$, we find that the similarity thresholds tend to be greater due to the condition that the elements in the tolerant set is contained in the same class and thus the classification result shows that the sizes of partitions become too small. Sometimes, the classification result leads to an extreme case that most partitions consist of only a single object. So, it is required that some objects that are contained in the same class are tolerable as much as possible in order to compensate this over-partitions. To meet the above second requirement, we define the ratio of good connections $\alpha_{\tau_A, \{d\}}$ that expresses a ratio of good connections to all possible connections as

$$\alpha_{\tau_A, \{d\}} = \frac{card(\tau_A \cap \{(x, y) | d(x) = d(y)\})}{card(\{(x, y) | d(x) = d(y)\})} \quad (11)$$

As the similarity threshold values increases, the ratio of good connections $\alpha_{\tau_A, \{d\}}$ decreases because the size of tolerant sets decreases with the increase of the value of similarity thresholds, and thus the number of good connections becomes smaller.

Since two coefficients $\gamma_{\tau_A, \{d\}}$ and $\alpha_{\tau_A, \{d\}}$ are operating inversely with the increase of the value of similarity thresholds, we take the fitness function F in order to balance two coefficients as

$$F = \omega \times \gamma_{\tau_A, \{d\}} + (1 - \omega) \times \alpha_{\tau_A, \{d\}}, \quad (12)$$

where ω and $(1 - \omega)$ are the weight constants that can be changed according to the goal of classification. Here, the first term makes some tolerant objects to be contained in the same class and the second term makes the objects in the same class to be tolerant.

3.4 Genetic Operations

The initial population of similarity thresholds are then evolved by appropriate genetic operations in order to find a set of optimal similarity thresholds for the pattern classification. The detailed explanation about the genetic operations used for the determination of the optimal

similarity thresholds is given as follows.

3.4.1 Reproduction

We use a mixture of selection methods [9] for reproducing the chromosomes. The first selection method is an elitism that the best chromosome with the highest fitness value is passed in the new population. The second selection method is a modified k -tournament method. In this method, a chromosome having the best fitness value among the k chromosomes selected from the upper class of fitness values randomly is chosen for the reproduction. Two chromosomes C_1 and C_2 obtained by repeating the above procedure consecutively create a new chromosome C_{c+m} by applying the crossover and mutation operations explained later. The above reproduction procedure is repeated as many times as $pselect \times |P|$, where $|P|$ is the population size. Finally, the remaining portion of the population set is filled by copying the population set in the order of magnitude of fitness values. Fig. 2 shows a hybrid reproduction method based on a mixture of three different reproduction methods.

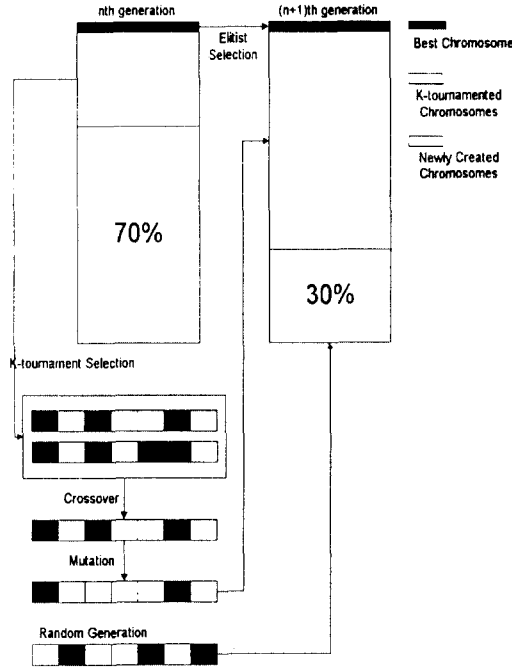


Figure 2: A proposed hybrid reproduction scheme.

3.4.2 Crossover

Basically, the crossover between two selected chromosomes C_1 and C_2 is performed as follow. Let the i th similarity threshold values of two selected chromosomes C_1 and C_2 be $t_1(a_i)$ and $t_2(a_i)$ and the fitness values of two selected chromosomes C_1 and C_2 be F_1 and F_2 . Then, the i th similarity threshold value $t_c(a_i)$ of the new chromosome C_c created by

the crossover operation is computed by an average weighted by fitness value as

$$t_c(a_i) = \frac{F_1 \times t_1(a_i) + F_2 \times t_2(a_i)}{F_1 + F_2} \quad (13)$$

This operation is applied to the overall similarity thresholds of two selected chromosomes in the crossover probability P_c .

3.4.3 Mutation

Mutation is performed as follows. Firstly, a chromosome C' is selected randomly from the population in the mutation probability P_m . Secondly, the similarity threshold value $t'(a_i)$ in the selected chromosome C' is randomly selected and it is mutated by the following.

$$t_m(a_i) = 1.5 - t'(a_i), \quad (14)$$

where $t'(a_i)$ and $t_m(a_i)$ are the selected similarity threshold and the mutated similarity threshold, respectively. The above computation implies that the mutation turns over the threshold value with respect to 1.5. The consecutive execution of the crossover and mutation operations complete the genetic operation and it creates the new chromosome C_{c+m} . Fig. 3 summarizes the determination of the optimal similarity threshold values using the genetic algorithm.

Algorithm (Input : $A = (U, A \cup d)$, $s_a : V_a \times V_a \rightarrow [0, 1] \forall a \in A$;
Output : $\{t \cup \{t(a) : a \in A\}\}$)

1. Initialization
 - Read information table.
 - Define the similarity measure.
 - Generate initial population : Take initial thresholds in $\{0, 1\}$.
 - Evaluate fitness function of initial population;
2. Perform the genetic algorithm
 - while \neg (stop.condition) {
 - Reproduction();
 - Crossover();
 - Mutation();
 - Evaluate fitness function of new population;
 - $\omega_1 \times \gamma_{r_s, \{d\}} + (1 - \omega_1) \times \alpha_{r_s, \{d\}}$
3. Determine the optimal similarity threshold values.

Figure 3: Determination of the optimal similarity threshold values using GA.

4 Pattern Classification based on the Tolerant Rough Set

We propose a new two-stage pattern classification method based on the lower and upper approximation set extracted from the set of training samples. Basically, the method is performed in the stage-wise in a way that the test sample are trying to be classified using the lower approximation set in the first stage and then the test sample that are impossible to classify in the previous stage are trying to be classified using the upper approximation set in the second stage. A detailed explanation of each stage's task is given below.

4.1 1st stage : Classification using the Lower Approximation Set

We obtain a tolerant set $TS_l(x)$ of a test sample x , where the subscript l denotes the lower approximation set. When the tolerant set $TS_l(x)$ of the test sample x consists of only itself, i.e. $TS_l(x) = \{x\}$, we can not classify the test sample x in the first stage and we pass the classification of such a test sample to the second stage. When a tolerant set $TS(x)$ of a test sample x consists of many training samples, the relative frequency of class inclusion of the training samples in the tolerant set $TS(x)$ is obtained. When the training samples in the tolerant set $TS(x)$ split the decisions, we assign the test sample x to the class that has the largest relative frequency value by the majority voting method. When the difference between the largest and the second largest relative frequency values is not too much, i.e., $\frac{freq_1 - freq_2}{freq_1} \leq \frac{1}{r(d)}$, where $freq_1$ and $freq_2$ are the largest and the second largest relative frequencies, respectively, and $r(d)$ is the number of decision classes, we also pass the classification of such a test sample to the second stage because the class inclusion is uncertain (fuzzy) under the the given lower approximation.

4.2 2nd stage : Classification using the Upper Approximation Set

Since the upper approximation set includes the all training samples in the lower approximation set and the training samples in the lower approximation set has been considered in the first stage, we are using the training samples in the boundary region, not all samples in the upper approximation set, in the second stage of classification. So, the computation time for classification in the second stage is not too long because only the samples in the boundary region are taken and the number of samples in the boundary region is not too many. Similarly in the first stage, we obtain a tolerant set $TS_b(x)$ of an unclassified test data x , where the subscript b means the boundary region. Then, we obtain the rough membership functions of all samples in the tolerant set $TS_b(x)$ of the unclassified test sample x with respect to the decision classes using Eq. (20). Let the tolerant set $TS_b(x)$ of the unclassified test sample x be $\{y_1, y_2, \dots, y_M\}$, where M is the number of samples in the boundary region that is tolerant with the test sample x , and assume that each tolerant sample y_j has the rough membership functions $\{\mu_d(y_j) | i = 1, 2, \dots, r(d)\}$. Then, we can compute the average rough membership function of the test sample x with respect to each decision class as

$$\bar{\mu}_d(x) = \frac{1}{M} \sum_{j=1}^M \mu_d(y_j), \quad i = 1, 2, \dots, r(d). \quad (15)$$

We assign the unclassified test sample x to the class that has the largest average rough membership function. When the difference between the largest and the second largest

average rough membership function is not too much, i.e., $\frac{\bar{\mu}_{d_{max1}} - \bar{\mu}_{d_{max2}}}{\bar{\mu}_{d_{max1}}} \leq \frac{1}{r(d)}$, where $\bar{\mu}_{d_{max1}}$ and $\bar{\mu}_{d_{max2}}$ are the largest and the second largest average rough membership function, respectively, and $r(d)$ is the number of decision classes, we reject the test sample x in the process of classification. Fig. 4 shows the procedure of our proposed two-stage classification method.

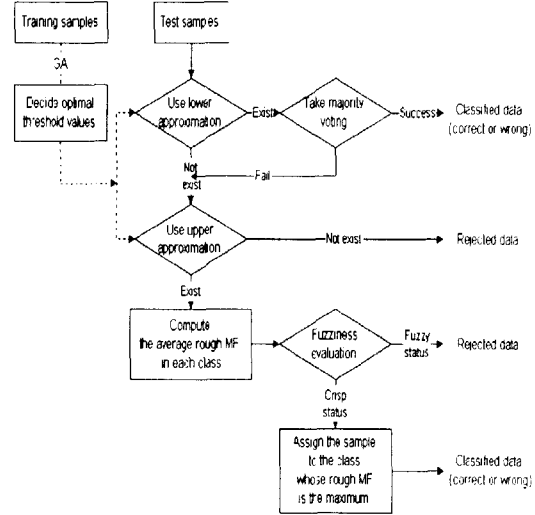


Figure 4: The proposed two-stage classification method.

5 Simulation Results and Discussion

We performed the pattern classification using the 100 handwritten numeral character set [10] in order to validate the proposed pattern classification method and compared the classification rate and training time with other classification methods. Fig. 5 shows a sample data set that consists of 100 handwritten numeral characters (ten sets of 10 numeral characters from 0 to 9). The 100 handwritten numeral characters are read by the 300 dpi HP scanner and they are normalized in the size of 18×18 pixels. Each numeral character is characterized by 20 attributes that is defined by the pixels of four nonoverlapping subregions (left, right, up, and down) and along the four diagonal half-cross axes (left-up, left-down, right-up, and right-down)[11].

We obtain 12 attributes ($a_1 - a_{12}$) by performing the vertical projection, the horizontal projection, and the number of pixels of four nonoverlapping subregions and obtain 8 attributes ($a_{13} - a_{20}$) by checking the existence of pixels and counting the number of pixels along the four diagonal half-cross axes. Table 1 shows the implication of each attribute, the range of the attribute value, and the quantization of the attribute value. Table 2 shows the attribute values of 50 sample data that are randomly selected 5 samples among the 10 samples per each class. We used the 50 sam-

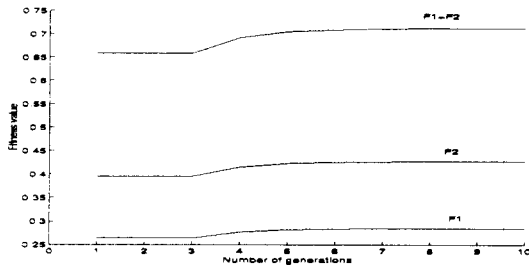


Figure 6: Evolution curves of various fitness functions.

network (simulation 2).

Table 5: Classification performances using all attributes (Simulation 1 and 2).

| Classification | Tolerant rough set | Neural network |
|-------------------|---------------------|---------------------|
| Confusion matrix | 5 0 0 0 0 0 0 0 0 0 | 5 0 0 0 0 0 0 0 0 0 |
| | 0 5 0 0 0 0 0 0 0 0 | 0 5 0 0 0 0 0 0 0 0 |
| | 0 0 5 0 0 0 0 0 0 0 | 0 0 5 0 0 0 0 0 0 0 |
| | 0 0 0 5 0 0 0 0 0 0 | 0 0 0 5 0 0 0 0 0 0 |
| | 0 0 0 0 5 0 0 0 0 0 | 0 0 0 0 5 0 0 0 0 0 |
| | 0 0 0 0 0 5 0 0 0 0 | 0 0 0 0 0 5 0 0 0 0 |
| | 0 0 0 0 0 0 5 0 0 0 | 0 0 0 0 0 0 5 0 0 0 |
| | 0 0 0 0 0 0 0 5 0 0 | 0 0 0 0 0 0 0 5 0 0 |
| | 0 0 0 0 0 0 0 0 5 0 | 0 0 0 0 0 0 0 0 5 0 |
| | 0 0 0 0 0 0 0 0 0 5 | 0 0 0 0 0 0 0 0 0 5 |
| Rejection | None | None |
| Misclassification | None | None |

Table 6 shows the comparison results of the classification performances of simulation 3 and 4 in terms of confusion matrix. where additional 50 sample data that were not included in the training phase were tested and each sample data was represented by all attributes. From this table, it is noted that there is one rejected sample (53th sample) and three misclassified samples (6th, 80th, and 84th sample) in the case of using the tolerant rough set (simulation 3) and there is no rejected sample and three misclassified samples (6th, 55th, and 80th sample) in the case of using the neural network (simulation 4). The notation 6th (0 → 9) in the fourth row of the Table 6 represents that the 6th sample whose true class is '0' is misclassified into the class '9'.

Table 6: Classification performances using all attributes (Simulation 3 and 4).

| Classification | Tolerant rough set | Neural network |
|-------------------|--------------------------------------|--------------------------------------|
| Confusion matrix | 9 0 0 0 0 0 0 0 0 1 | 9 0 0 0 0 0 0 0 0 1 |
| | 0 10 0 0 0 0 0 0 0 0 | 0 10 0 0 0 0 0 0 0 0 |
| | 0 0 10 0 0 0 0 0 0 0 | 0 0 10 0 0 0 0 0 0 0 |
| | 0 0 0 10 0 0 0 0 0 0 | 0 0 0 10 0 0 0 0 0 0 |
| | 0 0 0 0 9 0 0 0 0 0 | 0 0 0 0 10 0 0 0 0 0 |
| | 0 0 0 0 0 10 0 0 0 0 | 0 0 0 0 0 9 1 0 0 0 |
| | 0 0 0 0 0 0 10 0 0 0 | 0 0 0 0 0 0 10 0 0 0 |
| | 0 0 0 0 0 0 0 10 0 0 | 0 0 0 0 0 0 0 10 0 0 |
| | 0 0 0 0 0 0 0 1 0 8 1 | 1 0 0 0 0 0 0 0 0 9 0 |
| | 0 0 0 0 0 0 0 0 0 10 | 0 0 0 0 0 0 0 0 0 10 |
| Rejection | 53th | None |
| Misclassification | 6th(0 → 9), 80th(8 → 9), 84th(8 → 6) | 6th(0 → 9), 55th(5 → 6), 80th(8 → 0) |

Fig. 7 compares the classification perfor-

mances and the training times among four different simulations when each sample data is represented by all attributes. From this figure, it is noted that (1) the training time to be taken for determining the optimal similarity threshold values is very shorter than that for determining the weight vectors of the neural network and (2) the classification performances of two classification methods are almost similar in terms of the number of rejected and misclassified samples.

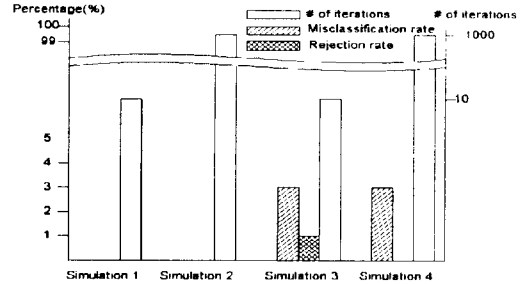


Figure 7: Comparison of classification performances and learning times using all attributes.

6 Conclusion

The proposed two-stage pattern classification method based on the tolerant rough set requires to define the similarity measure by the distance function of attributes between two objects, to determine the tolerant set among the objects, and to determine the lower and upper approximation set of the the objects based on the class information.

We used a very simple distance function like the absolute difference of attribute between two objects due to its low computational cost and We used the genetic algorithm to determine the optimal similarity threshold values, where the goal of evolution is to balance two requirements such that (1) it is required that some objects that are tolerant each other are included in the same class as many as possible and (2) it is required that some objects that are contained in the same class are tolerable as much as possible. After finding the optimal similarity threshold values, we computed the tolerance set of each training and sample. Based on the tolerance set, we determined the lower and upper approximation set of the training samples. The proposed pattern classification method consisted of two stages as follows. First, the test samples were classified using the lower approximation set. Next, the unclassified test samples in the first stage were re-classified using the rough membership values based on the upper approximation set. Some fuzzy test sample were rejected in the process of pattern classification.

Simulation results showed that (1) the number of misclassified samples in the case of using the tolerant rough set was slightly smaller than that in the case of using the neural network, which reflected more excellent avoid-

ing power of the classification by the tolerant rough set, (2) the number of rejected samples in the case of using the tolerant rough set was slightly greater than that in the case of using the neural network, which reflects the soft decision power of the classification by the tolerant rough set, and (3) the training times of two classification methods were $O(10)$ and $O(10^3)$ in terms of number of iterations in the case of using the tolerant rough set and the neural network, respectively, which reflects the very fast training time of the classification by the tolerant rough set.

We believe that the superiority of the classification performance of the proposed stage-wise pattern classification method comes from the fact the similarity threshold values that were required to determine the similarity between two objects in the training samples were determined optimally by using the genetic algorithm. One limitation of the proposed pattern classification method is that we need to know the class invocation of the training samples in advance. In future, we will extend the proposed pattern classification method based on the tolerant rough set to the unsupervised clustering problem.

References

- [1] G. Carpenter and S. Grossberg, "A Massively Parallel Architecture for a Self-organizing Neural Pattern Recognition Machine," *Computer Vision, Graphics, and Image Understanding*, vol. 37, pp. 54-115, 1987.
- [2] Chin-Teng Lin and C. S. George Lee, "Neural-network-based Fuzzy Logic Control and Decision System," *IEEE Transaction on Computers*, vol. 40, pp. 1320-1336, Dec. 1991.
- [3] P. Simpson, "Fuzzy Min-max Neural Networks-Part 1: Classification," *IEEE Trans. Neural Networks*, vol. 3, pp. 776-786, Sept. 1992.
- [4] Z. Pawlak, J. Grzymala-Busse, R. Slowinski, and W. Ziarko, "Rough Sets," *Communications of the ACM*, vol. 38, no. 11, pp. 89-95, Nov. 1995.
- [5] K. Funakoshi and T. B. Ho, "Information Retrieval by Rough Tolerance Relation," *The 4th International Workshop on Rough Sets, Fuzzy Sets, and Machine Discovery*, Tokyo, Nov. 6-8, 1996.
- [6] M. Kretowski and J. Stepaniuk, "Selection of Objects and Attributes : A Tolerance Rough Set Approach," *ICS Research Reports*, no. 21, Warsaw University of Technology, Warsaw, 1994.
- [7] J. H. Holland, *Adaptation in Natural and Artificial Systems*, Ann Arbor: The University of Michigan Press, 1992.
- [8] David E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Press, 1989.
- [9] Daijin Kim and Chulhyun Kim, "Forecasting Time series with Genetic Fuzzy Predictor Ensemble," *IEEE Trans. on Fuzzy Systems*, vol. 5, no. 4, pp. 523-535, 1997.
- [10] Y. Le Cun, "Handwritten Digit Recognition: Applications of Neural Net Chips and Automatic Learning," *Neurocomputing: Algorithms, Architectures and Applications*, Springer-Verlag, 1994.
- [11] D. Nejman, "Rough Set Approach to Handwritten Numeral Classification," *ICS Research Reports*, no. 18, Warsaw University of Technology, Warsaw, 1994.