

A Meta-Fuzzy Classifier for Specifying Appropriate Fuzzy Partitions by Genetic Fuzzy Rule Selection with Data Complexity Measures

Yusuke Nojima, Shinya Nishikawa, Hisao Ishibuchi

Dept. of Computer Science and Intelligent Systems, Osaka Prefecture University
1-1 Gakuen-cho, Naka-ku, Sakai, Osaka, 599-8531, Japan
{nojima@, nishikawa@ci., hisaoi@}cs.osakafu-u.ac.jp

Abstract— Tens of thousands of classifiers have been proposed so far. There is no best classifier among them for all the existing data sets. The performance of each classifier often depends on the data sets used for comparison. Even for a single classifier, suitable parameters of the classifier also depend on the data sets. That is, there is a possibility that a suited classifier and its parameter specification can be chosen beforehand if the target data sets or their characteristics were known. In recent years, a number of data complexity measures have been proposed to characterize data sets. The aim of this study is to develop a meta-classifier for selecting an appropriate classifier and/or its appropriate parameter specification by means of data complexity measures. In this paper, we focus on the parameter specification of fuzzy classifiers using data complexity measures as a preliminary study. To construct a meta-classifier, we generate a large number of artificial data sets from Keel benchmark data sets. Then we generate meta-patterns which are composed of the values of data complexity measures as inputs and an appropriate fuzzy partition as an output. Using meta-patterns, a meta-classifier is designed by multiobjective genetic fuzzy rule selection. We evaluate the proposed method through leave one-group out cross-validation.

Keywords—fuzzy classifier design; data complexity measures; meta-learning; pattern classification; knowledge acquisition

I. INTRODUCTION

A lot of comparative studies on pattern classification problems have shown that there is no single best classifier for all the existing data sets. In the practical use of classifiers, we do not know the best classifier for given data sets beforehand. We also do not know an appropriate parameter setting of the classifier used for the data sets. Thus, we usually select a classifier and specify its parameter values in a trial and error manner. This is a very time-consuming task. If we know a best-suited classifier and its appropriate parameter setting, we can reduce the redundant time.

In general, the performance of classifiers strongly depends on the data sets used in the experiments. For example, it would be enough to use a simple classifier for a simple data set. On the other hand, for a complicated data set, it must be necessary to use a complicated classifier which can finely approximate

the complicated decision boundaries. It is, however, difficult for us to know whether the data set is simple or complicated. To characterize the data sets, a number of data complexity measures for classification problems have been proposed by Ho and Basu [1,2]. Some researchers analyzed the relationship between the search performance of classifiers and data complexity measures [3-5]. Other researchers utilized data complexity measures to reduce training patterns [6-8].

In this paper, we focus on parameter specification of fuzzy classifiers by means of data complexity measures. We use a Chi-RW classifier which is a Chi's fuzzy classifier [9] with rule weights [10, 11]. The Chi-RW classifier is very famous and useful as a baseline of fuzzy classifiers. The main parameter of the Chi-RW classifiers is the number of fuzzy membership functions (i.e., membership granularity). Thus, we specify the membership granularity by utilizing data complexity measures before constructing fuzzy classifiers in this paper.

In our former study [12], we proposed a meta-classifier that predicts an appropriate granularity by the values of data complexity measures for each data set. While two-dimensional two-class problems were used in [12], we use a large number of data sets derived from Keel data repository in this paper. We propose *Leave one-group out* cross-validation for examining the prediction performance of meta-classifiers. A meta-classifier is obtained by multiobjective genetic fuzzy rule selection with genetic lateral tuning [13-16].

This paper is organized as follows. In Section II, we explain the overall procedure of the proposed framework. From Section III to Section VI, we explain the details on data preparation, data complexity measures, meta-data generation, and multi-objective genetic fuzzy rule selection. Section VII presents computational experiments. Finally, we conclude this paper in Section VIII.

II. OVERALL PROCEDURE OF THE PROPOSED FRAMEWORK

In this section, we briefly explain the overall procedure of the proposed framework by dividing it into two phases: training phase and predicting phase. The basic part of the proposed framework is the same as [12].

A. Training Phase

The procedure of the training phase is as follows:

This study is partially supported by the Japanese Grant-in-Aid for Young Scientists (B) (22700239).

- Step 1: A large number of artificial data sets are generated from the real world problems.
- Step 2: The values of data complexity measures are calculated for each data set.
- Step 3: The performance of a Chi-RW classifier with each candidate parameter setting is examined for each data set.
- Step 4: Meta-patterns are generated from the values of data complexity measures and the results on candidate parameter settings.
- Step 5: A meta-classifier is obtained by multiobjective genetic fuzzy rule selection using the meta-patterns.

Since existing data sets are limited and too few to cover the wide range of data complexity measures, we generate a large number of artificial data sets by eliminating some patterns from the original data sets, adding small noise into patterns, or generating new patterns nearby the original patterns in Step 1.

As a target classifier, we use a Chi-RW method [9-11] in this paper. It is very famous and well-used as a baseline. We focus on the granularity of fuzzy membership functions (i.e., the number of fuzzy membership functions). A single meta-pattern is corresponding to a single data set. The input vector of a meta-pattern is composed of the values of data complexity measures. The output class of the meta-pattern is the best-suited parameter setting in this paper.

A meta-data set with a large number of meta-patterns is used for designing a meta-classifier. In this paper, we employ multiobjective fuzzy rule selection with genetic lateral tuning [13-16].

B. Predicting Phase

The obtained meta-classifier in the training phase is used to predict an appropriate parameter setting of a classifier for unseen data sets. The procedure of the predicting phase is as follows:

- Step 1: The values of data complexity measures for an unseen data set are calculated.
- Step 2: The meta-classifier predicts class (i.e., an appropriate setting) according to the values of data complexity measures.

It should be noted that it would be possible to select a classifier among alternatives (e.g., RBFN, C4.5, SVM) instead of parameter settings for a Chi-RW classifier. It would be also possible to use other machine learning techniques for designing a meta-classifier.

The details are explained at the subsequent sections.

III. DATA PREPARATION

While we generated two-dimensional artificial data sets in [12], we generate multiple-dimensional artificial data sets from Keel data repository in this paper. Since each artificial data set is derived from one of the real-world data sets, each data structure is not completely meaningless.

To generate artificial data sets, we propose three simple operations. Figure 1 demonstrates a simple example. Each attribute value is normalized within $[0.0, 1.0]$.

1) Random reduction

Randomly remove a number of patterns as in Fig. 1 (b). A prespecified number of data sets are generated by changing the reduction rate.

2) Noise addition

Randomly add noise (e.g., uniform random value $[-0.05, 0.05]$ per variable) to each pattern as in Fig. 1 (c). Each new pattern is replaced with the original patterns. A prespecified number of data sets are generated by changing a random seed.

3) Random generation

Randomly generate new patterns in the neighbor of each pattern as in Fig. 1 (d). A prespecified number of data sets are generated by changing the number of newly-generated patterns.

If the new attribute value is less than 0.0 or more than 1.0, its value is replaced with 0.0 or 1.0, respectively.

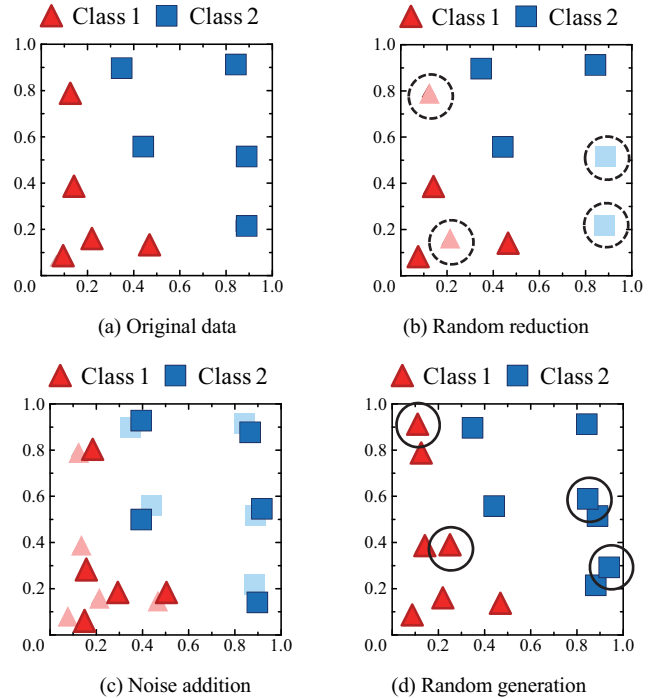


Figure 1. An example of artificial data set generation.

IV. DATA COMPLEXITY MEASURES

To describe the characteristics of data sets, this paper uses data complexity measures. In the literature [3-8], some researchers showed the effectiveness of data complexity measures. In this section, we briefly describe representative data complexity measures. Most of them have been originally defined for two-class data sets. The original explanations on the data complexity measures can be found in the literature (Ho and Basu [1, 2] and Orriols-Puig [17]).

A. F1: Maximum Fisher's discriminant ratio

This measure is calculated by the distance and variance between classes. It analyzes the effectiveness of a single attribute. The value is calculated as follows:

$$F1 = \max_{h=1,2,\dots,d} \frac{(\mu_{h1} - \mu_{h2})^2}{\sigma_{h1}^2 + \sigma_{h2}^2}, \quad (1)$$

where μ_{h1} , μ_{h2} , σ_{h1} and σ_{h2} are the means and the variances of two classes for the h -th attribute, respectively. The d is the number of attributes of the data set.

B. F2: Volume of overlap region

This measure describes how much two classes are overlapping each other, calculating the volume of the overlap region as hypercube of each dimension. This measure is strongly dependent on the number of attributes, because the value of an attribute's overlap region is multiplied as:

$$F2 = \prod_{h=1}^d \frac{\text{minmax}_h - \text{maxmin}_h}{\text{maxmax}_h - \text{minmin}_h}, \quad (2)$$

where

$$\begin{aligned} \text{minmax}_h &= \min\{\max(f_h, c_1), \max(f_h, c_2)\}, \\ \text{maxmin}_h &= \max\{\min(f_h, c_1), \min(f_h, c_2)\}, \\ \text{maxmax}_h &= \max\{\max(f_h, c_1), \max(f_h, c_2)\}, \\ \text{minmin}_h &= \min\{\min(f_h, c_1), \min(f_h, c_2)\}, \end{aligned}$$

where $\max(f_h, c_i)$ and $\min(f_h, c_i)$ are the maximum and minimum values of the h -th attribute in the class c_i ($i = 1, 2$), respectively (see Fig. 2).

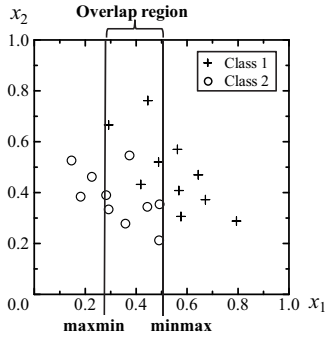


Figure 2. Overlap region of an attribute.

C. F3: Maximum (individual) feature efficiency

This measure describes the level of overlap like F2 measure. Different from F2, it counts the patterns which are falling outside of the overlap region for each attribute and considers the single attribute which has the maximum points. This value is calculated as follows:

$$F3 = \max_{h=1,2,\dots,d} \frac{|P_h|}{N_e}, \quad (3)$$

$$P_h = \{\mathbf{x}_i \mid x_{ih} \leq \text{maxmin}_h \text{ or } x_{ih} \geq \text{minmax}_h\}, \quad (4)$$

where N_e is the number of examples in the data set. x_{ih} is the h -th attribute value of the i -th pattern $\mathbf{x}_i$.

D. F4: Collective feature efficiency

The idea of this measure is derived from F3. In F3, only a single attribute is considered. However, F4 is calculated as follows. First, a single attribute is selected by (3). Then the overlap region on the attribute is calculated. The patterns out of the region are temporally removed. Next, another attribute is selected by (3). This process is repeated until all patterns are removed or all attributes are selected. This measure returns the fraction of all patterns by total removed patterns.

E. L1: The minimized sum of the error distance of a linear classifier.

This measure evaluates how much the training data are linearly separable. Therefore, its value is the sum of the difference between the exception of a linear classifier and the actual class value. Different from Ho and Basu [2], this paper uses a support vector machine (SVM) [18] trained with the sequential minimal optimization (SMO) algorithm [19] to construct the linear classifier. This measure can be used only for two-class problems. For a linearly separable problem, this measure becomes zero.

F. L2: Error rate of linear classifier

This measure is the error rate of the linear classifier designed for L1, measured with the training data set.

G. N1: Fraction of points on class boundary

This measure builds a class-blind minimum spanning tree (see Fig. 3) over the entire data set, and counts the number of points on an edge going across the different classes. This measure returns the fraction of these points over all points in the data set.

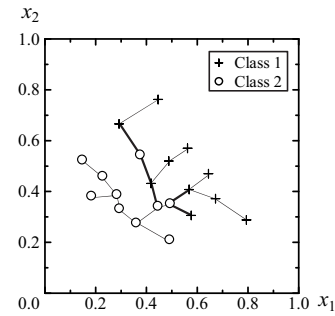


Figure 3. Minimum spanning tree. The bold lines are edges going across the different classes.

H. N2: The ratio of average intra/inter class nearest neighbor distance

This measure compares the spread with distance of within-class to that of the nearest neighbor of other classes. To compute this metric, we first compute the Euclidean distance to the nearest neighbor within the class and to the nearest

neighbor of other classes for all points ($\mathbf{x}_i$) in the data set. Then we compute the average of all the distance to intra class ($intraDist(\mathbf{x}_i)$) and the average of all the distance to inter class ($interDist(\mathbf{x}_i)$). This measure returns the ratio of these two averages. This measure is defined as follows:

$$N2 = \frac{\sum_{i=0}^{N_e} intraDist(\mathbf{x}_i)}{\sum_{i=0}^{N_e} interDist(\mathbf{x}_i)}. \quad (5)$$

I. N3: The leave-one-out error rate of 1-NN classifier

This metric simply returns the leave-one-out error rate by the 1-NN classifier.

J. L3: The nonlinearity of a linear classifier

This measure is characterized by a measure of non-linearity proposed by Hoekstra and Duin [20]. When a data set is given, a test set is created by linear interpolation with random coefficients between randomly selected pairs of points which are the same class. A linear classifier used for calculating L1 is also employed. After the classifier trained with the original data set, the error rate is measured by the created test set. This measure returns the error rate.

K. N4: The nonlinearity of 1-NN classifier

This measure creates a test set as proposed by L3, and evaluates the error rate of 1-NN. The value is the error rate.

L. T1: The fraction of maximum covering spheres

Simply speaking, this measure counts the number of balls needed to cover each class. At first, the ball is centered each instances and then grown as much as possible before touching the ball of other classes. The balls included any other ball are removed. This measure returns the fraction of all points by those balls.

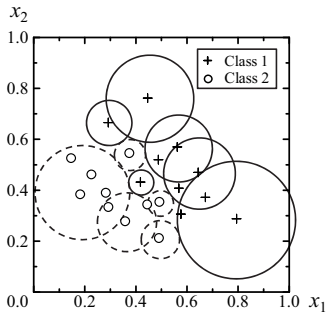


Figure 4. An example of the concept of T1 in the decision space.

Figure 4 shows the concept of T1, where the dashed circle means the balls of Class 2, the other circle means the balls of Class 1. In this example, the number of balls is 11.

M. T2: Average number of points per dimension

This measure represents the density of patterns.

$$T2 = \frac{N_e}{d}. \quad (6)$$

Since the calculation of all the data complexity measure is not computationally costly, we employ these values as inputs. We utilized DCoL (Data Complexity Library in C++) developed by Orriols-Puig et al [16] in order to calculate each data complexity measures.

V. A TARGET CLASSIFIER AND META-DATA GENERATION

In this section, we explain a target classifier (i.e., Chi-RW) and how to generate a meta-data set.

A. Target Classifier

In this paper, as the benchmark classifier, we use a Chi-RW classifier [9-11]. The main parameter of Chi-RW classifiers is the number of fuzzy membership functions (i.e., membership granularity). Thus, we focus on the granularity of fuzzy sets used in the antecedent of Chi-RW classifiers. We examine three different granularity specifications in Fig. 5: (a) coarse granularity (three fuzzy sets), (b) moderate granularity (five fuzzy sets), and (c) fine granularity (seven fuzzy sets). We call the Chi-RW classifiers with those granularities Chi-RW-3, Chi-RW-5, and Chi-RW-7, respectively.

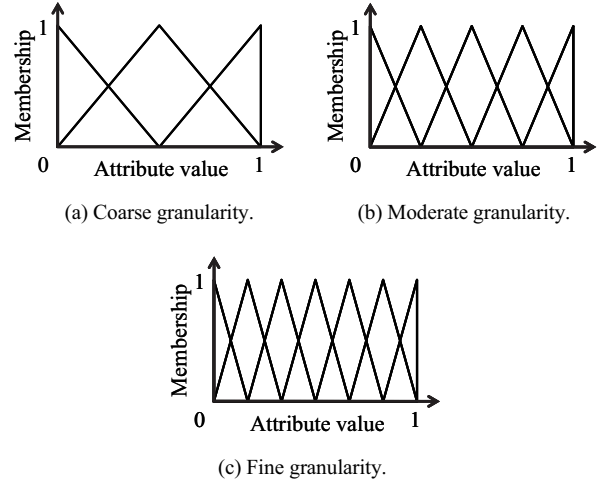


Figure 5. Fuzzy membership functions used in Chi-RW classifiers.

As explained in introduction, the objective of this paper is to estimate an appropriate granularity of fuzzy membership functions in Chi-RW classifiers for each data set. It can be simply expected that a coarse granularity must be suited for data sets with noisy patterns. On the other hand, a fine granularity must be suited for data sets with complex decision boundaries. For some other data sets, a moderate granularity might be useful.

B. Meta-data generation

We generate a meta-data set composed of a large number of meta-patterns from many data sets. We assume that an appropriate granularity is strongly dependent on each data set. So that, we define each meta-pattern corresponding to one data

set. The input values of each meta-pattern are the values of data complexity measures of a data set explained in Section IV. Let $\mathbf{z}_i = (z_{i1}, z_{i2}, \dots, z_{ij}, \dots, z_{in})$, $i = 1, 2, \dots, D$ be the attribute values (i.e., the values of the data complexity measures) of i -th data set, where D is the number of data sets. We use 13 data complexity measures: F1, F2, F3, F4, N1, N2, N3, N4, L1, L2, L3, T1 and T2 for two-class data sets in this paper (i.e., $n = 13$). For two or more-class data sets, we use nine data complexity measures: F2, F3, F4, N1, N2, N3, N4, T1 and T2. It should be noted that all input values (i.e., the values of data complexity measures) are normalized. This normalization is just to use the same membership functions for each attribute. Since we do not know the actual minimum and maximum values of each data complexity measure, we should examine the effect of this normalization. This is one of our future research issues.

The output class of each meta-pattern is an appropriate granularity of fuzzy membership functions in a Chi-RW classifier for the data set. Let $y_i \in \{\text{Chi-RW-3, Chi-RW-5, Chi-RW-7}\}$ be the consequent class of i -th meta-pattern. For each meta-pattern, three Chi-RW classifiers are performed by a ten-fold cross-validation (10CV). The classifier with the best test data accuracy is chosen as the consequent class of the meta-pattern. We performed 10CV by means of the KEEL data-mining software tool [21].

VI. META-CLASSIFIER DESIGN BY MULTIOBJECTIVE GENETIC FUZZY RULE SELECTION

From the generated meta-data set, we obtain a meta-classifier that predicts an appropriate fuzzy membership functions. We apply multiobjective genetic fuzzy rule selection (MoGFRS) to the design of a meta-classifier [13-16]. Genetic fuzzy rule selection [22] is one of the well-known methods in the field of genetic fuzzy systems [23,24]. It is a very simple two-stage method [22]. At the first stage, a number of fuzzy if-then rules are generated from numerical data as candidate rules. At the second stage, a subset of candidate rules is chosen by a genetic algorithm with binary strings in which “0” and “1” mean “not selected” and “selected” as a classifier, respectively. By maximizing the classification performance of the classifier and minimizing the complexity of the classifier (e.g., the number of fuzzy rules), an accurate and simple rule set is obtained as the final classifier. Its multiobjective formulation versions have been also studied in [13]. We use the modified version [15] of MoGFRS. There are two modifications: One is the use of a single granularity specification for each attribute [14]. The other is the use of genetic lateral tuning for adjusting fuzzy membership functions [16].

The procedure of our rule selection method is as follows:

- Step 1: Rule extraction with multiple granularities.
- Step 2: Specification of a single granularity for each attribute based on the extracted rules.
- Step 3: Rule extraction with selected single granularities.
- Step 4: Multiobjective genetic fuzzy rule selection.

In Step 1, a prespecified number of fuzzy rules of the following type are extracted from the meta-patterns.

$$R_j: \text{If } z_1 \text{ is } A_{j1} \text{ and } \dots \text{ and } z_n \text{ is } A_{jn} \text{ then Class } y_j \text{ with } CF_j, \quad (7)$$

where R_j is the label of the j -th fuzzy rule, $\mathbf{z} = (z_1, \dots, z_n)$ is an n -dimensional meta-pattern vector, A_{ji} is an antecedent fuzzy set such as “Simple” and “Complicated”, y_j is a consequent class, and CF_j is a rule weight. We use 14 fuzzy membership functions shown in Fig. 6 and a *don't care* condition as the antecedent fuzzy set. To obtain interpretable knowledge, we restricted the number of conditions to three or less fuzzy sets.

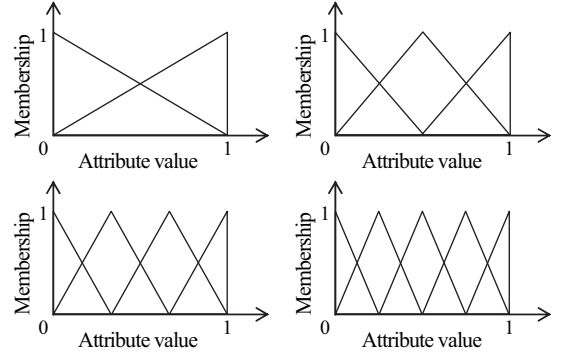


Figure 6. Membership functions for a meta-classifier design.

After extracting the rules, the frequently-used promising granularity is selected for each attribute in Step 2. In Step 3, a prespecified number of fuzzy rules are extracted by using the selected single granularity for each attribute. In Step 4, the combination of extracted rules and the positions of fuzzy membership functions are optimized by NSGA-II of Deb et al. [25].

We use a double coding scheme [15] in our multiobjective genetic fuzzy rule selection with genetic lateral tuning (Fig. 7). Each string includes binary bits and real numbers. Binary bits represent whether each extracted rule is selected for a meta-classifier, or not. Real numbers represent the position of each fuzzy membership function as shown in Fig. 8. The movable range is limited to maintain the interpretability of the fuzzy membership functions.

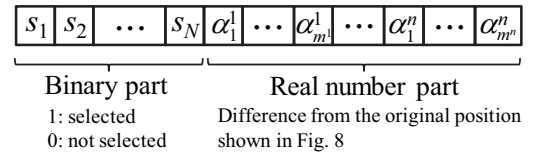


Figure 7. String representation by the double coding scheme. m'' is the total number of linguistic variables (fuzzy membership functions) for n -th input.

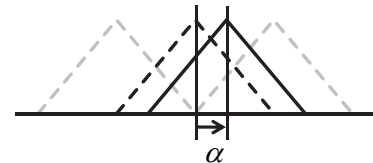


Figure 8. Genetic lateral tuning of fuzzy membership functions.

Each meta-classifier S is evaluated by the following three objective functions.

$f_1(S)$: the number of correctly classified training meta-patterns,

$f_2(S)$: the number of selected fuzzy rules,

$f_3(S)$: the total number of antecedent conditions.

That is, our multiobjective genetic fuzzy rule selection problem is formulated as:

$$\text{Maximize } f_1(S) \text{ and minimize } f_2(S) \text{ and } f_3(S). \quad (8)$$

Uniform crossover and bit-flip mutation are used as genetic operations for the rule selection part.

BLX- α crossover [26] and uniform mutation are used as genetic operations for tuning part.

By our multiobjective genetic fuzzy rule selection, we obtain a number of non-dominated decision systems with respect to the three objective functions. We select the most accurate classifier with respect to the first objective $f_1(S)$ for the training meta-data set, as the final meta-classifier.

VII. COMPUTATIONAL EXPERIMENTS

We examined two cases: One case is the application of the proposed framework to two-class problems. The other case is the application to multiple class problems with two or more classes. We explain the experimental setting and show the experimental results in this section.

A. Experimental Setting

We generated 100 artificial data sets from each data set by each simple operation explained in Section III. Thus, 300 new artificial data sets were generated in total. However, if more than one Chi-RW classifiers with different granularity have the same classification performance, we eliminate such data sets from this experiment.

The parameter settings of data set generation are as follows:

Reduction rate (random reduction): 5%, 10%, ..., 50%,

Generation rate (random generation): 10%, 20%, ..., 100%.

The parameter settings of MoGFRS are as follows:

The number of candidate rules per class: 300,

The number of strings: 200,

The number of generations: 5000,

Crossover probability: 0.9,

Mutation probability: 0.1 (1 to 0), $1/N$ (0 to 1),

(N is the number of total candidate rules).

Different mutation probabilities are used to select a small number of rules from the candidate ones.

To examine the generalization ability of the obtained meta-classifier, we propose *Leave one-group out* cross-validation shown in Fig. 9. Unlike 10-fold cross-validation (10CV), data sets are not randomly subdivided. All the data sets derived

from an original data set are regarded as a group. When some group is used as test data sets, the other groups are used as training data sets. This is because each data set in the same group has somewhat similar data structure to each other. We avoided this affect in our experiments in order to assume the practical use of our proposed framework. As one of future issues, we have to examine whether the generated artificial data sets can cover the input space for unseen data sets or not.

We used the following experimental procedures.

- Step 1: Divide artificial data sets into training data sets (T_r) and test data sets (T_s) by leave one-group out cross-validation.
- Step 2: Calculate the values of data complexity measures of each artificial data set of T_r .
- Step 3: Classify each artificial data set of T_r using each one of three granularities by 10CV.
- Step 4: Generate a meta-data set from Step 2 and Step 3.
- Step 5: Obtain a meta-classifier by MoGFRS.
- Step 6: Calculate the values of data complexity measures of each artificial data set of T_s .
- Step 7: Predict an appropriate granularity for each data set.
- Step 8: Classify each data set of T_s using the granularity predicted in Step 7.

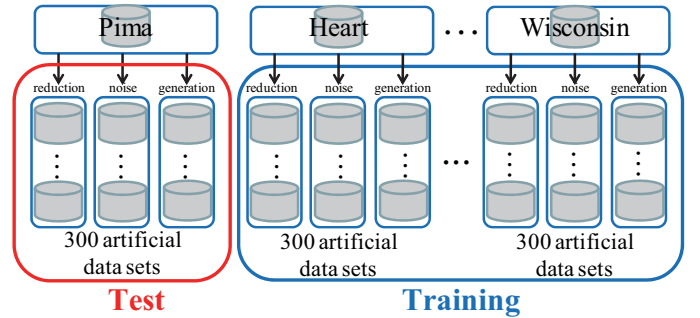


Figure 9. Leave one-group out validation.

B. Experimental Results for Two-class Problems

As two-class problems, we used 13 data sets listed in Table I. They are available from Keel data repository (<http://keel.es>). By simple operations, we generated 3,900 data sets in total.

The classification rates for test patterns in 10CV are shown in Table II. Chi-RW-3, Chi-RW-5, and Chi-RW-7 represent the results obtained by Chi-RW classifiers with coarse, moderate, and fine granularity, respectively. In those cases, the fixed granularity of fuzzy membership functions was used for all the data sets. "Meta-classifier" represents the result obtained by Chi-RW classifiers in which a granularity was selected by the meta-classifier for each data set. "Best" represents the result obtained by Chi-RW classifiers in which the best granularity was always used for each data set. This is an ideal case and just for a reference. From Table II, we can see that better test data

accuracy was obtained by our proposed method than the classifiers with fixed granularity. Table II also indicates that the result by our proposed method was close to the ideal case.

TABLE I. USED DATA SETS WITH TWO CLASSES

Data sets	Patterns	Attributes	Classes
Appendicitis	106	7	2
Australian	690	14	2
Banana	5300	2	2
Bupa	345	6	2
Haberman	306	3	2
Heart	270	13	2
Mammographic	830	5	2
Monk-2	432	6	2
Pima	768	8	2
Sonar	208	60	2
Titanic	2201	3	2
Wdbc	569	30	2
Wisconsin	683	9	2

TABLE II. CLASSIFICATION RATE OF TWO-CLASS PROBLEMS

	Chi-RW-3	Chi-RW-5	Chi-RW-7	Meta-Classifier	Best
Classification Rate (%)	75.11	65.89	59.77	77.18	77.84

The classification rate for test group data sets in *Leave one-group out* cross-validation was 77.00%.

We also analyzed if-then rules of the obtained meta-classifier. Since there were large differences among the meta-classifiers by leave one-group out, we generated a meta-classifier using all the artificial data sets (i.e., there is no test group). Figure 10 shows the best classifier with respect to the first objective $f_1(S)$. It was composed of four fuzzy rules. The black triangles mean the selected fuzzy membership functions. The left end of the x-axis means “simple”, while the right end of the x-axis means “complicated”.

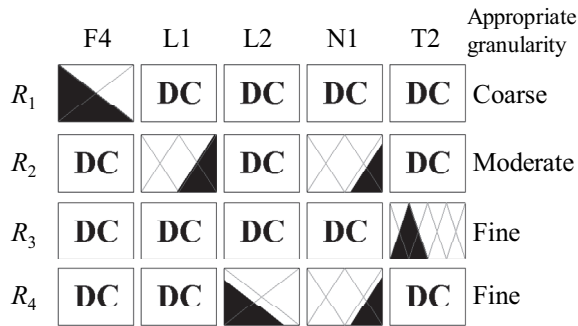


Figure 10. The obtained meta-classifier for two-class problems by MoGFRS.

Figure 10 can be interpreted as follows:

R_1 : If the data set is **simple** w.r.t. **F4** then **coarse** fuzzy set is the best.

R_2 : If the data set is **complicated** w.r.t. **L1** and **N1** then **moderate** fuzzy set is the best.

R_3 : If the data set is **simple** w.r.t. **T2** then **fine** fuzzy set is the best.

R_4 : If the data set is **simple** w.r.t. **L2** and it is **complicated** w.r.t. **N1** then **fine** fuzzy set is the best.

These rules are counterintuitive except for R_1 .

C. Experimental Results for multiple class Problems

As multiple class problems with two or more classes, we used 24 data sets listed in Table III. They are also available from Keel data repository (<http://keel.es>). By simple operations, we generated 7,200 data sets in total.

Table IV shows the classification rates for test patterns in 10CV. From Table IV, we can see that better test data accuracy was obtained by our proposed method than the classifiers with fixed granularity. But the difference between our proposed method and Chi-RW-3 seems to be very small.

TABLE III. USED DATA SETS WITH TWO AND MORE CLASSES

Data sets	Patterns	Attributes	Classes
Appendicitis	106	7	2
Australian	690	14	2
Bands	365	19	2
Bupa	345	6	2
Cleveland	297	13	5
Dermatology	358	34	6
Glass	214	9	6
Haberman	306	3	2
Hayes-roth	132	4	3
Heart	270	13	2
Hepatitis	80	19	2
Ionosphere	351	34	2
Iris	150	4	3
Mammographic	830	5	2
Newthyroid	215	5	3
Pasture	36	22	3
Pima	768	8	2
Saheart	462	9	2
Sonar	208	60	2
Tae	151	5	3
Vehicle	846	18	4
Wdbc	569	30	2
Wine	178	13	3
Wisconsin	683	9	2

TABLE IV. CLASSIFICATION RATE OF MULTI-CLASS PROBLEMS

	Chi-RW-3	Chi-RW-5	Chi-RW-7	Meta-Classifier	Best
Classification Rate (%)	75.63	64.25	55.47	75.72	77.02

It should be noted that the ratio of data sets for which Chi-RW-3 was the best among three Chi-RW classifiers was 70.37%. Since the classification rate for test group data sets in *Leave one-group out* cross-validation was 70.83%, we can say

that the meta-classifier successfully selected different granularity to some extent.

Figure 11 shows the best classifier with respect to the first objective $f_1(S)$. No rule with moderate class was included in the meta-classifier. N4 measure seems to be a key data complexity measure for this case.

	F2	F3	N2	N3	N4	T2	Appropriate granularity
R_1		DC		DC	DC	DC	Coarse
R_2	DC	DC	DC				Coarse
R_3		DC		DC		DC	Fine
R_4	DC			DC		DC	Fine

Figure 11. The obtained meta-classifier for multi-class problems by MoGFRS.

VIII. CONCLUSIONS

We proposed a framework for specifying an appropriate parameter setting in the design of Chi-RW classifiers for unseen data sets. The experimental results showed that our proposed framework can correctly predict an appropriate granularity by data complexity measures. The results by our proposed method were better than those by using the same (fixed) granularity for all data sets.

As a future research topic, we will apply this framework to classifier selection in which an appropriate classifier is selected from popular machine learning techniques (e.g., RBFN, SVM, C4.5) by data complexity measures for each data set. It would be necessary to reconsider the artificial data generation in order to generate data sets with various values of data complexity measures. A recent proposal [27] could be an alternative method.

REFERENCES

- [1] T. K. Ho and M. Basu, "Complexity measures of supervised classification problems," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, March 2002, pp. 289-300.
- [2] T. K. Ho, M. Basu, and M. H. Law, "Measures of geometrical complexity in classification problems," M. Basu and T. K. Ho (eds.) *Data Complexity in Pattern Recognition*, Springer-Verlag, 2006, pp. 3-23.
- [3] S. Singh, "Multiresolution estimates of classification complexity," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, no. 12, December 2003, pp. 1534-1539.
- [4] J. Luengo and F. Herrera, "Domains of competence of fuzzy rule based classification systems with data complexity measures: A case of study using a fuzzy hybrid genetic based machine learning method," *Fuzzy Sets and Systems*, vol. 161, no. 1, 2010, pp. 3-19.
- [5] A. Orriols-Puig and J. Casillas, "Fuzzy knowledge representation study for incremental learning in data streams and classification problems," *Soft Computing*, (in press).

- [6] J. S. Sanchez, R. A. Mollineda, and J. M. Sotoca, "An analysis of how training data complexity affects the nearest neighbor classifiers," *Pattern Analysis and Application*, vol. 10, no. 3, 2007, pp. 189-201.
- [7] R. A. Mollineda, J. S. Sanchez, and J. M. Sotoca, "Data characterization for effective prototype selection," *Proc. of the 2nd Iberian Conf. on Pattern Recognition and Image Analysis*, 2005, pp. 27-34.
- [8] S. Garcia, J. R. Cano, E. Bernado-Mansilla, and F. Herrera, "Diagnose effective evolutionary prototype selection using an overlapping measure," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 23, no. 8, 2009, pp. 1527-1548.
- [9] Z. Chi, H. Yan, and T. Pham, *Fuzzy Algorithms: with Applications to Image Processing and Pattern Recognition*, World Scientific, 1996.
- [10] O. Cordon, M. J. del Jesus, and F. Herrera, "A proposal on reasoning methods in fuzzy rule-based classification systems," *International Journal of Approximate Reasoning*, vol. 20, no. 1, 1999, pp. 21-45.
- [11] H. Ishibuchi and T. Yamamoto, "Rule weight specification in fuzzy rule-based classification systems," *IEEE Trans. on Fuzzy Systems*, vol. 13, no. 4, 2005, pp. 428-435.
- [12] S. Nishikawa, Y. Nojima, and H. Ishibuchi, "Appropriate granularity specification for fuzzy classifier design by data complexity measures," *Proceedings of the World Congress on Nature and Biologically Inspired Computing*, Kitakyusyu, Japan, December 15-17, 2010, pp. 698-703.
- [13] H. Ishibuchi and T. Yamamoto, "Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining," *Fuzzy Sets and Systems* vol. 141, 2004, pp. 59-88.
- [14] R. Alcalá, Y. Nojima, F. Herrera, and H. Ishibuchi, "Generating single granularity-based fuzzy classification rules for multiobjective genetic fuzzy rule selection," *Proc. of the 2009 IEEE International Conference on Fuzzy Systems*, Jeju, Korea, pp. 1718-1723.
- [15] R. Alcalá, Y. Nojima, F. Herrera, and H. Ishibuchi, "Multiobjective genetic fuzzy rule selection of single granularity-based fuzzy classification rules and its interaction with the lateral tuning of membership functions," *Soft Computing* (in press).
- [16] R. Alcalá, J. Alcalá-Fdez, and F. Herrera, "A Proposal for the genetic lateral tuning of linguistic fuzzy systems and its interaction with rule selection," *IEEE Transactions on Fuzzy Systems*, vol. 15, no. 4, 2007, pp. 616-635.
- [17] A. Orriols-Puig, N. Macia, and T. K. Ho, "Documentation for the data complexity library in C++," La Salle - Universitat Ramon Llull, Quatre Camins 2, Barcelona, Spain, 2010001, 2010. <http://dcol.sourceforge.net>
- [18] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer Verlag, New York, 1995.
- [19] C. Platt, "Fast training of support vector machines using sequential minimal optimization," *Advances in Kernel Methods: Support Vector Learning*, MIT Press, Cambridge, MA, USA, 1998, pp. 185-208.
- [20] A. Hoekstra and R. P. W. Duin, "On the nonlinearity of pattern classifiers," *Proc. of the 13th International Conference on Pattern Recognition*, Vienna, August, 1996, pp. 271-275.
- [21] KEEL A Software Tool to Assess Evolutionally Algorithms for Data Mining Problems. <http://keel.es/>
- [22] H. Ishibuchi, K. Nozaki, N. Yamamoto, and H. Tanaka, "Selecting fuzzy if-then rules for classification problems using genetic algorithms," *IEEE Trans. on Fuzzy Systems*, vol. 3, no. 3, 1995, pp. 260-270.
- [23] O. Cordon, F. Herrera, F. Hoffman, and L. Magdalena, *Genetic Fuzzy Systems*, World Scientific, 2001.
- [24] F. Herrera, "Genetic fuzzy systems: Taxonomy, current research trends and prospects," *Evolutionary Intelligence*, vol. 1, 2008, pp. 27-46, 2008.
- [25] Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. on Evolutionary Computation*, vol. 6, no. 2, April 2002, pp. 182-197.
- [26] L. J. Eshelman, K. E. Mathias, and J. D. Schaffer, "Crossover operator biases: Exploiting the population distribution," *Proc of 7th International Conference on Genetic Algorithms*, 1997, pp. 354-361.
- [27] N. Macia, A. Orriols-Puig, and E. Bernado-Mansilla, "In search of targeted-complexity problems," *Proc. of Genetic and Evolutionary Computation Conference*, 2010, pp. 1055-1062.