The 20th World Congress of the International Fuzzy Systems Association (IFSA 2023)
August 20~23, 2023 / EXCO, Daegu, Korea

Evaluation of Bagging-type Ensemble Method Generating Virtual Data

Honoka Irie† and Isao Hayashi‡
Graduate School of Informatics, Kansai University, Takatsuki, Osaka, Japan,
† rsuinaixiang@gmail.com
‡ ihaya@kansai-u.ac.jp

***Abstract*—For pattern classification problems, an ensemble-learning method identifies multiple weak classifiers using learning data and combines them to improve the discrimination rate of testing data. We have previously proposed possibilistic data interpolation bagging (pdi-Bagging), which improved the discrimination rate of testing data by adding virtually generated data to the learning data. However, the accuracy of the correct virtual data type is unstable because the virtual data are generated over a wide area of the data space. In addition, the discriminant accuracy is low because the evaluation index for changing the generation class of the virtual data is defined in each dimension. In this study, we propose a new method for specifying the generation area of virtual data and changing the generation class of the virtual data. Consequently, the discriminant accuracy improved because five new bagging methods that generate virtual data around the correct discrimination data and error discrimination data are formulated, and the class of virtual data is determined using the proposed new evaluation index in multidimensional space. We formulate the new pdi-Bagging algorithm and discuss the usefulness of the proposed method using numerical examples.**

***Index Terms*—Fuzzy Inference, Virtual Data, Ensemble Method, Bagging, Clustering**

I. INTRODUCTION

Recently, ensemble learning methods [1]–[3], which are useful for pattern classification problems, have been proposed. The ensemble method learns multiple weak classifiers through training data and can improve the classification accuracy of the evaluation data by combining multiple weak classifiers across the layers. Ensemble learning can be broadly categorized into two types: classifier and attribute combination models [4]. The classifier combination model combines weak classifiers, whereas the attribute combination model constructs weak classifiers with highly correlated attributes for the class patterns. The classifier combination model can be classified into two types: independent and dependent. In the independent type, each classifier is combined independently, whereas in the dependent type, each classifier is combined while maintaining a dependency relationship. In the independent type, each classifier is trained using individual training data. Thus, it is possible to integrate them independently and achieve a high classification rate. The independent type includes the bagging method [5], random forests [6], [7], and error-correcting output codes [8]. The bagging method represents bootstrap aggregation. The learning data for the classifier are obtained via bootstrap sampling, and multiple classifiers were learned independently from the learning data. The final result is obtained based on a majority vote involving all the integrated classifiers. Because the bagging method is a simple ensemble method that uses multiple classifiers, the algorithm is simple and offers high applicability. For example, it is often used as a clustering model for medical data [5] and a prediction model for time series [9]. In addition, it obtained a higher accuracy than AdaBoost when used as a model for detecting defects in semiconductor wafers [10]. In another case, it was applied to the Social Stratification and Mobility survey in Japan [11].

In contrast, there are boosting methods [12]–[14] and adaptive mixture methods of local experts [15] as the dependent type of classifier combination model. Boosting is a method for improving the classification rate by sequentially learning weak classifiers. AdaBoost [12] is particularly useful, and it has the advantage of being easy to analyze dataset features. Thus, the dependent type, represented by boosting, is trained using multiple weak classifiers while maintaining sequential interdependence with the training data and can identify the input–output relationship associated with the dependence. Contrastingly, for the independent type represented by bagging, the weak classifier is independent for each training dataset. However, the processing algorithm is relatively simple and highly accurate. In contrast, an ensemble method that integrates bagging and boosting has also been proposed [16].

We have proposed a new bagging algorithm for the generation and interpolation of data around misclassified data using a specified membership function [17]–[19]. We call this method possibilistic data interpolation bagging (pdi-Bagging). The interpolation of data around misclassified data is called virtual data. In pdi-Bagging, data misclassified by the classifier model are not weighted, as in AdaBoost, nor are they added to the subsequent training data. The classes of the virtual data are estimated using locations [20], [21], and the virtual data are added to the training data to estimate the discriminant lines using the weak classifiers based on fuzzy inference [22]–[24]. Similarly, in the next layer, the class of virtual data is estimated and added to the training data to estimate the discriminant line. This series of operations is repeated, and the classification rate of the evaluation data is obtained using a majority vote of multiple weak classifiers. As the amount of data increases with the addition of virtual data during training, the amount of data in each class is equalized by eliminating the bias in the amount of data between classes, which improves the accuracy of the

discriminant line identification. In this study, we formulate a new pdi-Bagging algorithm and discuss the usefulness of this method using numerical examples. Specifically, we formulate five types of virtual data generation methods and discuss their usefulness. We also discuss the usefulness of a new evaluation index that changes the output class of the virtual data.

II. PDI-BAGGING

The pdi-Bagging method identifies multiple weak classifiers through training data, and the final class output of the checking data is determined by a majority rule using a plurality of weak classifiers. A conceptual diagram of pdi-Bagging is shown in Fig. 1. In pdi-Bagging, the weak classifiers M_0 of the fuzzy inference are learned using training data that is probabilistically extracted from all datasets, and the discriminant rate of the training data TRD is calculated. Subsequently, virtual data are generated around the misclassified data using membership functions. The generated virtual data are added to the original training data to increase the amount of training data TRD. Using the original training data and virtual data, the classification rate is calculated by a weak classifier M_1 based on fuzzy inference. Increasing the number of TRD improves the discriminant accuracy of weak classifiers. The repetition of operations is completed L times when the end judgment is satisfied. Finally, the evaluation data (CHD) are input into L weak classifiers $M_0, M_1, \cdots, M_l, \cdots, M_L$, and the final result is then calculated using the majority rule. Since pdi-Bagging adds virtual data to the training data and calculates the discriminant rate using multiple weak classifiers, its discriminant rate was higher than that of the conventional bagging method and AdaBoost [17], [18].

In pdi-Bagging, fuzzy clustering using simple fuzzy inference [22] is adopted as the weak classifier. Fuzzy inference has excellent learning abilities and can visualize learning results using rule descriptions. Therefore, fuzzy inference is adopted as a weak classifier. Simplified fuzzy inference expresses rules in if–then form, uses fuzzy sets defined by the membership functions in the antecedent part, and defines the consequent part in singleton form using real numbers. Here, we used a trapezoidal fuzzy set as the membership function.

Let z be the output variable, and p_i be a singleton in the consequent part, the fuzzy rule, r_i, $i = 1, 2, \cdots, R$, is expressed as follows:

$$r_i \quad : \quad \text{if } x_1 \text{ is } \mu_{F_{i1}}(x_1) \text{ and } \cdots \text{ and } x_n \text{ is } \mu_{F_{in}}(x_n) \\ \text{then } C = \{C_{ik} \mid z = p_i\}$$

where C is the output class, and C_{ik} indicates that the class value is C_k for rule r_i.

Suppose we obtain the input data $\mathbf{x} = (x_1, x_2, \cdots, x_n)$. The input data $\mathbf{x}$ are input into the antecedent part of the i-th fuzzy rule r_i, and the degree of the antecedent part, $\mu_i(\mathbf{x}) = \mu_{F_{i1}}(x_1) \cdot \mu_{F_{i2}}(x_2) \cdot \cdots \cdot \mu_{F_{in}}(x_n)$, is calculated. The results of the fuzzy inference, $\hat{z}$, and class C are calculated using the following equations:

$$\hat{z} = \frac{\sum_{i=1}^{R} \mu_i(\mathbf{x}) \cdot p_i}{\sum_{i=1}^{R} \mu_i(\mathbf{x})}$$
$$C = \{C_k \mid \min |\hat{z} - z|\}$$

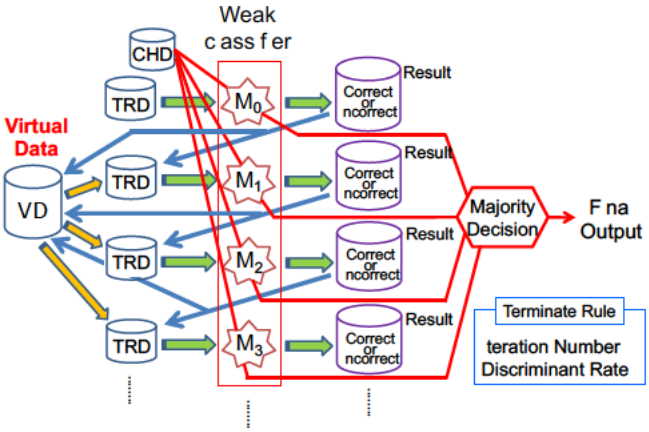


Fig. 1: pdi-Bagging Algorithm

Let us now explain the generation of virtual data using pdi-Bagging. Let $\mathbf{x}^D(d) = (x_1^D(d), x_2^D(d), \cdots, x_j^D(d), \cdots, x_n^D(d))$ denote the d-th data in the dataset D consisting of W data points. The virtual data $\mathbf{x}^V(d)$ are generated around correctly discriminated data(correct-classified data) $\mathbf{x}^C(d)$ and misclassified data $\mathbf{x}^E(d)$. For a certain real number, h, $0 \le h \le 1$, the virtual data $x_j^V(d)$ of the j-th attribute of $\mathbf{x}^V(d)$ is generated using the membership function $\mu_F(x_j)$ of the fuzzy number F as follows:

$$x_j^V(d) = \{x_j \mid \mu_F(x_j) = h, \mu_F(x_j^S(d)) = 1\}$$
$$h \sim N(1,1), \quad 0 \le h \le 1$$

where $x_j^S(d)$ denotes the correct-classified data $x_j^C(d)$ or the misclassified data $x_j^E(d)$. In addition, the membership function $\mu_F(x_j)$ is defined as the following normal distribution, whose center is $x_j^S(d)$ and whose standard deviation is σ.

$$\mu_F(x_j) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x_j - x_j^S(d))^2}{2\sigma^2}\right) \quad (1)$$

We propose the following five methods for generating virtual data:

(1) CA: Virtual data generation method with correct classified data in the whole space
When the training data $\mathbf{x}^S(d)$ is correctly classified using a weak classifier, virtual data $\mathbf{x}^V(d)$ are generated around the correct classifying data $\mathbf{x}^C(d)$.

(2) CC: Virtual data generation method with correct classified data at the cluster center
When the training data $\mathbf{x}^S(d)$ is misclassified by the weak classifier, the midpoint between the closest correct classified data and the farthest correct classified data from $\mathbf{x}^E(d)$, whose classes are the same as the

misclassified data, $\mathbf{x}^E(d)$ is calculated. Virtual data $\mathbf{x}^V(d')$ are generated around the correct classified data $\mathbf{x}^C(d')$ closest to the midpoint.

$$\mathbf{x}^C(d') = \{\mathbf{x}^C(e) \mid \min_e |\mathbf{x}^C(e) - \frac{1}{2}(\max_f |\mathbf{x}^E(d) - \mathbf{x}^C(f)| + \min_g |\mathbf{x}^E(d) - \mathbf{x}^C(g)|)|, \quad for\ \forall e, f, g\}$$

(3) E: Virtual data generation method with misclassified data

When the training data $\mathbf{x}^S(d)$ is discriminated as misclassified by the weak classifier, virtual data $\mathbf{x}^V(d)$ are generated around the misclassified data $\mathbf{x}^E(d)$.

(4) MA: Virtual data generation method by mixing correct classified data and misclassified data in the whole space

By alternately using the CA and E type in each bagging layer, virtual data $\mathbf{x}^V(d)$ is generated around $\mathbf{x}^C(d)$ and $\mathbf{x}^E(d)$.

(5) MC: Virtual data generation method by mixing correct classified data and misclassified data at the cluster center

By alternately using CC type and E type in each bagging layer, virtual data $\mathbf{x}^V(d)$ are generated around $\mathbf{x}^C(d)$ and $\mathbf{x}^E(d)$.

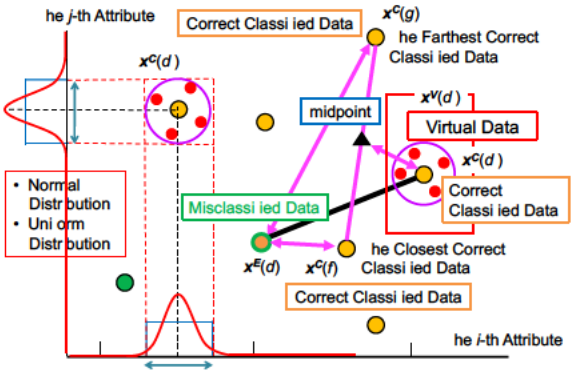


Fig. 2: Generation of Virtual Data with Correct Data Around Cluster Center

In particular, regarding CC, we explain how virtual data are generated around the correct classified data at the cluster center using Fig. 2. We assume that the clustering problem of a total of eight data points into two classes, green and yellow, is shown in Fig. 2. The training data with a green frame in yellow, located at the bottom of the figure was misclassified as the green class instead of the yellow class. Because the true class of the misclassified data $\mathbf{x}^E(d)$ is the yellow class, the midpoint between the closest and the farthest correct classified data from $\mathbf{x}^E(d)$, whose classes were yellow, is calculated. Virtual data $\mathbf{x}^V(d')$ are generated around the correct classified data $\mathbf{x}^C(d')$ whose class is yellow and closest to the midpoint. According to the generation method, several virtual data points in the CC method tended to be generated near the center of the cluster. Therefore, we should note that the generation of virtual data by the CC method tends to affect the discriminant line more than the CA method, which generates virtual data in the entire space. Additionally, it is possible to control the degree of influence on the discriminant line by moving the coordinate position currently set as the midpoint to an arbitrary interpolation or extrapolation point from the endpoints.

III. FORMULATION FOR CLASS MODIFICATION

In pdi-Bagging, virtual data are generated around misclassified data and correct classified data to improve the discriminant rate. However, to further improve the classification rate, it is necessary to assign the correct classes to the virtual data. Here, we propose a new class determination method for assigning the correct classes to the virtual data.

Suppose that virtual data $\mathbf{x}^V(d)$ are generated from the correct classified data $\mathbf{x}^C(d)$ and the misclassified data $\mathbf{x}^E(d)$. The class of the virtual data $\mathbf{x}^V(d)$ should be the same as the output class of the source data $\mathbf{x}^S(d) = \{\mathbf{x}^{C,k}(d),\ \mathbf{x}^{E,k}(d)\}$. However, virtual data may be generated at locations distant from the source data. In addition, virtual data may be generated in areas where different classes of data are dense. Therefore, the class k^* of the virtual data $\mathbf{x}^V(d)$ is determined by the integration evaluation formula using the following three evaluation criteria. Therefore, the class k^* of the virtual data $\mathbf{x}^V(d)$ is determined by the integration evaluation formula using the following three evaluation criteria: the evaluation of the correct/misclassified data (E_1), the evaluation of the class centers (E_2), the evaluation of neighborhood data classes (E_3).

(1) Evaluation of Correct/Misclassified Data(E_1)

The evaluation value E_1 is defined as the distance between virtual data $\mathbf{x}^V(d)$, source data $\mathbf{x}^{S,k}(d)$ with class k. The smaller the evaluation value E_1, the higher the dependence of $\mathbf{x}^V(d)$ on the class k.

$$E_1^k = \frac{|\mathbf{x}^V(d) - \mathbf{x}^{S,k}(d)|}{\max_e |\mathbf{x}^{S,k}(d) - \mathbf{x}^{D+V}(e)| - \min_f |\mathbf{x}^{S,k}(d) - \mathbf{x}^{D+V}(f)|}, \quad for\ \forall e, f$$

$$E_1^p = 1 - E_1^k,\ for\ p \neq k$$

(2) Evaluation of Class Centers(E_2)

The evaluation value E_2 is defined as the distance between virtual data $\mathbf{x}^V(d)$ and the center of class k. The smaller the evaluation value E_2, the higher the dependence of $\mathbf{x}^V(d)$ on the class k. Now, when the center of class k is represented as $\mathbf{x}_c^k$,

$$E_2^k = \frac{|\mathbf{x}^V(d) - \mathbf{x}_c^k|}{\max_{e,f} |\mathbf{x}^{D+V}(e) - \mathbf{x}^{D+V}(f)|},\ for\ \forall e, f$$

(3) Evaluation of Neighborhood Data Classes(E_3)

The evaluation value E_3 is defined as the distance between virtual data $\mathbf{x}^V(d)$ and the closest correct/misclassified data

$\mathbf{x}^{S,k}(e)$ with class k. The smaller the evaluation value E_3, the higher the dependence of $\mathbf{x}^V(d)$ on the class k.

$$E_3^k = \frac{\min_e |\mathbf{x}^V(d) - \mathbf{x}^{S,k}(e)|}{\max_{f,g} |\mathbf{x}^{D+V}(f) - \mathbf{x}^{D+V}(g)|}, \quad for \ \forall e, f, g$$

According to these three criteria, the evaluation E_1 is higher when virtual data are generated near the source data. In contrast, the evaluation E_2 is high when the virtual data generate near the center of the class.

By integrating these three evaluation criteria, the overall evaluation value E^k was obtained. The virtual data $\mathbf{x}^V(d)$ have a class k^* that minimizes the following overall evaluation value E^k.

$$k^* = \{k | \min_k E^k = \min_k (w_1 E_1^k + w_2 E_2^k + w_3 E_3^k)\} \qquad (2)$$

where w_1, w_2, w_3 denote the weights for each evaluation value.

We formulate the pdi-Bagging algorithm as follows:

Step 1 It is assumed that W and D data are obtained. Data D are categorized into two types of datasets: W^{TRD} training data D^{TRD} and W^{CHD} check data D^{CHD}. Therefore, $W = W^{TRD} + W^{CHD}$. In addition, interpolated data are represented by D^V.

Step 2 The training data D^{TRD} are used as inputs to the l-th weak classifier M_l, and the discriminant rate r_l^{TRD} is obtained. where M_0 denotes the initial weak classifier.

Step 3 The d-th data points that were correctly or misclassified were temporarily extracted from D^{TRD}. It is assumed that the d-th data point is misclassified. For the j-th attribute value $x_j^S(d)$ of the correct classified data or the misclassified data, virtual data $x_j^V(d)$ are generated by the membership function, $\mu_F(x_j)$.

Step 4 Calculate the class k^* of the virtual data $\mathbf{x}^V(d)$ using the equation (2). Remove the virtual data $\mathbf{x}^V(d)$ from the $l-1$th D^V with $l > 2$, and add the virtual data $\mathbf{x}^V(d)$ with class k to the lth D^V.

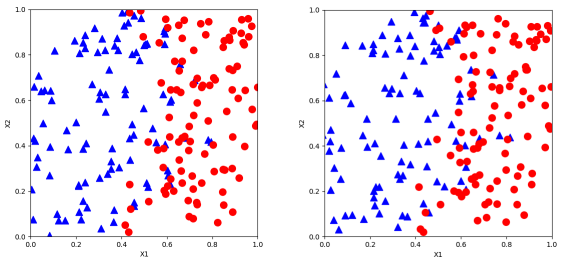
Step 5 Extract v pieces of virtual data from D^V using a random number and add them to D^{TRD}.

Step 6 Steps 2–4 are repeated with $l = l+1$, and the algorithm is terminated at $K = l$, satisfying $r_l^{CHD} \geq \theta$ for threshold θ. Alternatively, the algorithm ends when $l \geq K$ is satisfied for the number of weak classifiers L and the number of iterations K, $K \leq L$.

Step 7 To obtain the final discrimination result, D^{CHD} is applied to $M_0, M_1, \cdots, M_l, \cdots, M_K$, and then the discriminant rate r_K^{CHD} is obtained by majority rule.

IV. VERIFICATION AND DISCUSSION USING NUMERICAL DATA

To explain the pdi-Bagging algorithm, we discussed the two-dimensional classification problem. It is assumed that 200 training data points and 200 checking data points exist in a two-dimensional space of the interval [0,1] and that these data can be categorized into two classes. Fig. 3 shows the numerical data used as training and checking data. These numerical data were constructed by adding the value ± 0.05 to the basic data using random numbers. We deal with two-input and two-class discrimination problems as numerical data. For this discriminant problem, the real value of the consequent part of the fuzzy inference rules was set to 2.0 (red, ◯) and 3.0 (blue, △).



(a) Numerical Example Data 1 (b) Numerical Example Data 2

Fig. 3: Numerical Example Training and Testing Data

The simplified fuzzy inference was used as the weak classifier, and five types of trapezoidal membership functions were set for each input interval [0, 1]. Because the data space is two-dimensional, 25 rules were constructed over the entire space. Additionally, to verify the classification rate when the rules were added to the data space as specific areas, 49 rules were added to $G_1 = \{(x_1, x_2) \mid [0.4, 0.7] \times [0.4, 0.7]\}$ as the specific area G_1, and four rules were added to $G_2 = \{(x_1, x_2) \mid [0.7, 0.8] \times [0.3, 0.7]\}$ as the specific area G_2. Consequently, the total number of rules was 78. The addition of rules improved the discriminant accuracy rate in regions away from the discriminant line, where the data were dense, and the overall discriminant rate was improved. The discriminant rate was calculated for the following three types: no additional rule, membership function set in the trapezoidal shape, and membership function defined in the right-angled trapezoidal shape at both ends of specific regions. When the membership function in a specific region was set to right-angled trapezoid type at both ends of the specific region, the size of the specific region did not change, even if the membership functions were learned. However, when the trapezoidal membership functions were set at both ends of specific regions, the size of the region changed as the membership functions were learned. Therefore, when the right-angled trapezoidal membership function was set in the additional rules, the membership functions did not move outside the specific region, even when the membership functions were learned, which were intensively learned within a specific region.

The initial value of the antecedent part of fuzzy reasoning was set by the default method, and the learning order of the antecedent and consequent parts was that the consequent part was learned first, and then the antecedent and consequent parts were alternately learned. During the learning process, the learning coefficients of the x-coordinates x_b and x_c of the two vertices of the upper bases of the trapezoidal membership function denoted K_b and K_c and were set to 0.01 [24]. In addition, the learning coefficients of the difference α and β between the x-coordinates of the upper and lower bases denote K_α and K_β and were set to 0.01 [24]. However, the learning coefficient K_p of the singleton of the consequent part was set to 0.4 for the first consequent learning and 0.6 for the alternate learning. The number of epochs of the consequent part was set to 10, and the alternating learning of the consequent part was set to (10, 10).

As a membership function $\mu_F(x_j)$ for generating the virtual data, the normal distribution of Equation (1) with a standard deviation of $\sigma = 0.5$ was selected, and the number of virtual data generated was one. However, in preliminary experiments, the discriminant rate of the fuzzy inference was approximately 87%. As a result, approximately 26 of the 200 checking data points were erroneously classified, and approximately eight virtual data points were required to bring the total number of virtual data points to 200 training data points. Therefore, we discuss the discriminant rate when the number of generated virtual data varied from 1 to 10.

The evaluation value weights for class estimation of the virtual data were $(w_1, w_2, w_3) = \{(1/3,\ 1/3,\ 1/3),\ (0.2,\ 0.4,\ 0.4),\ (0.2,\ 0.3,\ 0.5),\ (0.2,\ 0.5,\ 0.3),\ (0.5,\ 0.25,\ 0.25),\ (0.01,\ 0.495,\ 0.495)$, and $(0.05,\ 0.475,\ 0.475)\}$. The weight w_1 of the distance from the source data significantly affects the class estimation in determining the weight. Therefore, we discussed the discriminant rate for a total of seven types: $w_1 = 1/3$ when $w_1 = w_2 = w_3$, $w_1 = 0.5$, and five types with the value of w_1 reduced.

The algorithm is terminated by the termination rule whose number of iterations is $K = 5$. In the mixed discriminant type, the type for the misclassified data was adopted in the odd layers, and the type for the correct classified data was adopted in the even layers. In the fuzzy-inference learning process, the order of the data is changed using random numbers every epoch. Because the number of epochs required for learning the consequent part and the alternate learning of the antecedent and consequent parts were 10 and (10, 10), respectively, the total number of epochs was 150 in Five-Layer Learning. As two-fold cross-validation was used here, 150 epochs of epoch learning were performed for each dataset, resulting in 300 epochs of learning. We compared the average discriminant rates obtained in ten trials for each type: CA, CC, E, MA, and MC.

The discriminant rate for evaluation data using five types of virtual data generation methods—the type of correct classified data in the whole space (CA), the type of correct classified data at the cluster center (CC), the type of misclassified data (E), the mixing type of correct classified and misclassified data in the entire space (MA), and the mixing type of correct classified and misclassified data in cluster center (MC)—are listed in Table I and Figures 4–6. Table I shows the discriminant rate for each weight with respect to the evaluation index, with and without additional rules, and with respect to the membership-function shape within a specific region. Additionally, we calculated the difference between the discriminant rate when 25 rules were set using a trapezoidal membership function. Figs. 4–6 show the average discriminant rate for the weight with respect to the evaluation index, with and without additional rules, and with respect to the membership-function shape within a specific region.

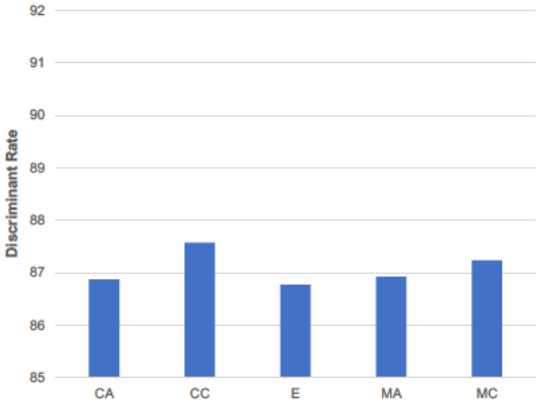


Fig. 4: Average Discriminant Rates of Five Methods with 25 Basic Rules

From the results presented in Table I and Fig. 4, the following characteristics of the discriminant rate are clear for the case of 25 rules with trapezoidal membership functions: The discriminant rate by two-fold cross-validation of fuzzy inference with 25 rules was 84.40%. The discriminant rate of all five methods that generate virtual data is higher than the result of this fuzzy inference. Therefore, the generation of virtual data is effective in improving the discriminant rate.

For the 25 rules of the trapezoidal membership function, the discriminant rate is not necessarily high. However, the discriminant rates of the five methods were higher than those of the 25 rules. For the types of correct classified data, the discriminant rate of CC is higher than that of CA, and even for the mixing types of correct classified data, the discriminant rate of MC is higher than that of MA. This is because, in CC and MC, virtual data are generated near the center of the cluster. Thus, the fuzzy rules near the center of the class were learned with high accuracy.

Table I and Fig. 5 show the characteristics of the discriminant rate for the 78 rules added within the specific region using a trapezoidal membership function. The discriminant rate of two-fold cross-validation of simple fuzzy inference with 78 rules using the trapezoidal membership function was 89.68%. However, of the five types of virtual data generation methods, the discriminant rates of the CC, E, and MC were higher than those of the simple fuzzy inference. Therefore, methods other than generating virtual data in the entire space are effective.

TABLE I: Comparison of Discriminant Rates Based on Five Methods

Rule Format	Evaluation Values Weight	CA (%)			CC (%)			E (%)			MA (%)			MC (%)		
		Dis.R.	Dif. (a)	Dif. (b)	Dis.R.	Dif. (a)	Dif. (b)	Dis.R.	Dif. (a)	Dif. (b)	Dis.R.	Dif. (a)	Dif. (b)	Dis.R.	Dif. (a)	Dif. (b)
(a) Trap.M.F. 25 Rules	1/3, 1/3, 1/3	86.73	—	—	87.70	—	—	86.61	—	—	87.05	—	—	87.29	—	—
	0.2, 0.4, 0.4	86.50	—	—	87.60	—	—	87.03	—	—	87.28	—	—	87.52	—	—
	0.2, 0.3, 0.5	87.00	—	—	87.55	—	—	86.70	—	—	87.03	—	—	87.10	—	—
	0.2, 0.5, 0.3	86.85	—	—	87.70	—	—	86.70	—	—	87.08	—	—	87.15	—	—
	0.5, 0.25, 0.25	86.40	—	—	87.45	—	—	86.95	—	—	86.85	—	—	87.40	—	—
	0.01, 0.495, 0.495	87.18	—	—	87.55	—	—	86.55	—	—	86.58	—	—	86.88	—	—
	0.05, 0.475, 0.475	87.45	—	—	87.48	—	—	86.85	—	—	86.63	—	—	87.30	—	—
	Average	86.87	—	—	87.58	—	—	86.77	—	—	86.93	—	—	87.23	—	—
(b) Trap.M.F. 78 Rules	1/3, 1/3, 1/3	89.53	2.80	—	89.83	2.13	—	89.80	3.18	—	89.78	2.73	—	89.79	2.50	—
	0.2, 0.4, 0.4	89.33	2.83	—	89.93	2.33	—	90.15	3.13	—	90.00	2.73	—	89.95	2.43	—
	0.2, 0.3, 0.5	89.03	2.03	—	90.15	2.60	—	90.30	3.60	—	89.78	2.75	—	89.93	2.83	—
	0.2, 0.5, 0.3	88.95	2.10	—	89.65	1.95	—	90.05	3.35	—	89.85	2.78	—	89.83	2.67	—
	0.5, 0.25, 0.25	89.18	2.77	—	89.48	2.03	—	90.05	3.10	—	89.38	2.53	—	90.23	2.83	—
	0.01, 0.475, 0.475	87.40	0.22	—	89.80	2.25	—	88.63	2.08	—	88.55	1.97	—	89.43	2.55	—
	0.05, 0.475, 0.475	88.70	1.25	—	90.00	2.52	—	89.83	2.97	—	89.85	3.23	—	89.85	2.55	—
	Average	88.87	2.00	—	89.83	2.26	—	89.83	3.06	—	89.60	2.67	—	89.86	2.62	—
(c) R.A.Trap.M.F. 78 Rules	1/3, 1/3, 1/3	90.03	3.30	0.50	90.33	2.63	0.50	89.93	3.32	0.14	90.23	3.18	0.45	90.15	2.86	0.36
	0.2, 0.4, 0.4	89.83	3.33	0.50	90.20	2.60	0.27	90.35	3.33	0.20	90.28	3.00	0.27	90.28	2.76	0.32
	0.2, 0.3, 0.5	90.45	3.45	1.43	90.10	2.55	-0.05	90.05	3.35	-0.25	90.10	3.08	0.32	90.30	3.20	0.37
	0.2, 0.5, 0.3	89.95	3.10	1.00	90.35	2.65	0.70	90.05	3.35	0.00	90.30	3.23	0.45	89.98	2.82	0.15
	0.5, 0.25, 0.25	90.18	3.78	1.00	90.28	2.83	0.80	89.93	2.97	-0.13	90.05	3.20	0.67	90.35	2.95	0.13
	0.01, 0.475, 0.475	87.55	0.37	0.15	90.40	2.85	0.60	88.63	2.07	0.00	88.40	1.82	-0.15	90.18	3.30	0.75
	0.05, 0.475, 0.475	89.83	2.37	1.13	90.35	2.87	0.35	89.95	3.10	0.12	90.03	3.40	0.17	90.03	2.73	0.17
	Average	89.69	2.81	0.81	90.29	2.71	0.45	89.84	3.07	0.01	89.91	2.99	0.31	90.18	2.94	0.32

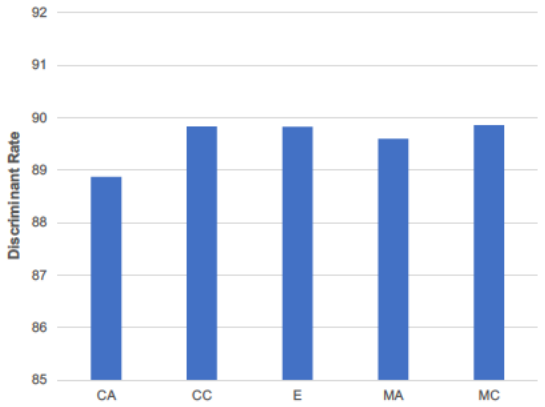


Fig. 5: Average Discrimination Rates of Five Methods with 78 Total Rules Added by Trapezoidal Membership Function

In addition, Table I and Fig. 6 show the characteristics of the discriminant rate for the 78 rules added within the specific region using the right-angled trapezoidal membership function. The discriminant rate of two-fold cross-validation of simple fuzzy inference with 78 rules, using the right-angled trapezoidal membership function, was 89.73%. Among the five types of virtual data generation methods, the discriminant rates of the four types, CC, E, MA, and MC, were higher than simple fuzzy inference. In particular, MC and CC were higher than 0.45%. Therefore, in the case of 78 rules with right-angled trapezoidal membership functions, the average discriminant rate was high for CC and MC. Based on the differences in the discriminant rates of the 25 rules of the trapezoidal membership function, the average discriminant rate increased by 2.71% to 3.07% for all five methods. However, the rate of increase in the average discriminant rate of CA and CC was slightly lower than those of the other methods. In addition, the average discriminant rate of the 78 rules for the right-angled trapezoidal membership function is 0.38% higher than that of the 78 rules of the trapezoidal membership function. In contrast, the maximum discriminant rate was 90.35% for CC when the weights of the evaluation index were (0.2, 0.5, and 0.3) and MC when the weights of the evaluation index were (0.5, 0.25, and 0.25). In a specific area, there are many singular data points; therefore, learning the rules in this area increases the overall discriminant rate. Additionally, when the right-angled trapezoidal membership functions were set in this specific region, the size of the specific region did not change. Thus, the membership functions are efficiently learned within the specific region, and the overall discriminant rate increases.

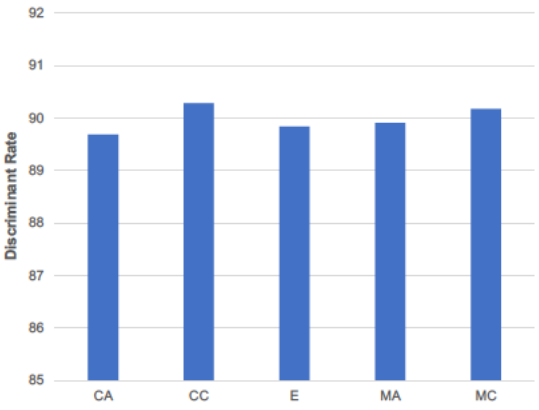


Fig. 6: Average Discrimination Rates of Five Methods with 78 Total Rules Added by Right Trapezoidal Membership Function

Table II shows the results of the *t*-test of the discriminant

TABLE II: Results of t-Test between Five Methods with 25 Basic Rules

Virtual Data Generation Method	CA	CC	E	MA	MC
CA	—	① 0.1779	① ②	① ②	① ②
CC	① 0.1779	—	① ②	① ②	② 0.0291
E	① ②	① ②	—	② 0.1978	① ②
MA	① ②	① ②	② 0.1978	—	① 0.2106
MC	① ②	② 0.0291	① ②	① 0.2106	—

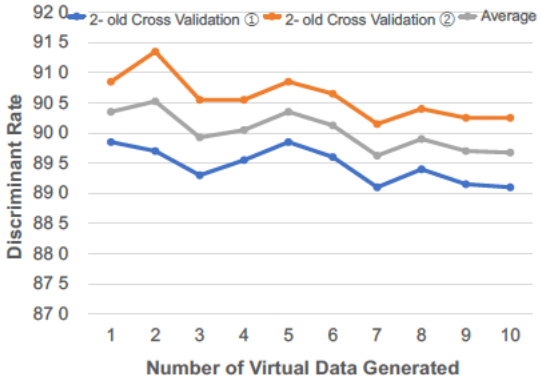


Fig. 7: Discriminant Rates Owing to Changes in Virtual Data in CC

Next, we discuss how the discriminant rate changes depending on the amount of virtual data generated. The maximum discriminant rate was 90.35% for 78 rules with right-angled trapezoidal membership functions. They were for CC when the weights of the evaluation index were (0.2, 0.5, 0.3) and MC when the weights of the evaluation index were (0.5, 0.25, 0.25). Therefore, we discuss the discriminant rate based on the amount of virtual data generated for the two weights. Fig. 7 and Fig. 8 show the changes in the discriminant rate. Figure 7 shows the discriminant rate and the average discriminant rate when the amount of virtual data changes from one to ten in the CC, with the weights of the evaluation index being (0.2, 0.5, and 0.3). In addition, Fig. 8 shows the discriminant rate and average value of the discriminant rate with respect to changes in the number of virtual data points in MC, with the weights of the evaluation index being (0.5, 0.25, and 0.25).

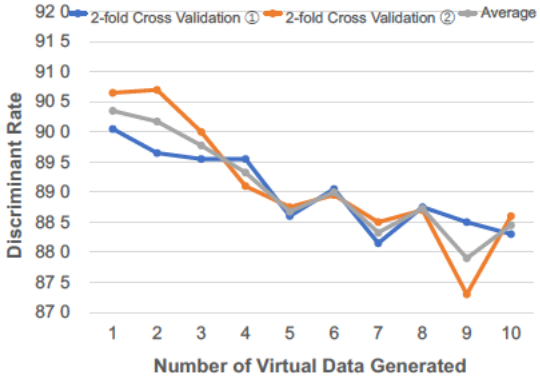


Fig. 8: Discriminant Rates Owing to Changes in Virtual Data in MC

In Fig. 7, the two types of discriminant rates using two-fold cross-validation gradually decrease while maintaining a constant difference without crossing, even when the amount of virtual data increases and the variance value is small. By contrast, the average discriminant rate peaked at the maximum discriminant rate of 90.53% when the amount of virtual data was two and then gradually decreased as the number of generated virtual data increased, and its variance value was small. The virtual data generated by the CC did not depend on the position of the misclassified data and were generated near the center of the cluster, where many correct classified data exist. Therefore, the discriminant rate was not affected by the amount of virtual data and exhibited almost the same values.

On the other hand, in Fig. 8, the discriminant rate decreased sharply as the number of generated virtual data increased. In particular, the two types of discriminant rate by two-fold cross-validation decreased sharply as the amount of virtual data increased with frequent crossings, and the variance value was large. By contrast, the average discriminant rate peaked at its maximum of 90.35% when the amount of virtual data was one and then decreased sharply as the number of generated virtual data increased. The minimum discriminant rate was 87.90% when the number of virtual data was 9. The difference in the discriminant rate was 2.45%. The virtual data by MC was generated near the correct classified and misclassified data. Therefore, the discriminant rate is strongly affected by the amount of virtual data generated, and the optimal number of generations of virtual data.

In summary, the methods with the highest discriminant rates were CC and MC, with 78 rules using the right-angled trapezoidal membership functions in specific regions. In both methods, the discriminant rate was improved by adding rules to specific regions where singularity data existed. In addition, as the membership function was defined by a right-angled

trapezoid, the specific region was not expanded, and the membership function was learned intensively. These reasons led to high discriminant rates. Next, we discuss the relationship between the number of virtual data generated and the discriminant rate. In CC, virtual data are generated from the correct classified data near the center of the cluster, but the number of the correct classified data points is large. Therefore, the discriminant rate is relatively constant without being affected by the number of virtual data points. In contrast, in MC, virtual data are generated from the correct classified data near the center of the cluster and the misclassified data in the entire space. Thus, the discriminant rate depends relatively on the number of generated virtual data. Therefore, after determining the weight of the evaluation index, the maximum discriminant rate can be obtained from CC and MC with the number of virtual data generated as a parameter. In this numerical example, the maximum discriminant rate was 90.53% for CC when the number of virtual data generated was two.

V. CONCLUSIONS

In this paper, we considered a method for generating virtual data and a method for changing classes using pdi-Bagging. In addition, we evaluated the accuracy of the generation method of virtual data and the class change using numerical examples.

In the future, we will investigate how virtual data can be generated when there is a bias in the amount of data between classes, and how to generate virtual data with directionality. Additionally, it is necessary to determine the usefulness of pdi-Bagging in practical applications using actual measurement data.

ACKNOWLEDGEMENTS

This work was partially supported by JST SPRING, Grant Number JPMJSP2150. In addition, this work was partly supported by JSTS KAKENHI Grant Number JP20K11981 of Grants-in-Aid for Scientific Research(C). This work was partly supported by the Kansai University Fund for Supporting Outlay Research Centers and the Kansai University Fund for Domestic and Overseas Research Fund.

REFERENCES

[1] R.Polikar: Ensemble Based Systems in Decision Making, *IEEE Circuits and Systems Magazine*, Vol.6, No.3, pp.21–45 (2006).

[2] L.Rokach: Taxonomy for Characterizing Ensemble Methods in Classification Tasks: A Review and Annotated ibliography, *Computational Statistics & Data Analysis*, Vol.53, No.12, pp.4046-4072, DOI:10.1016/j.csda.2009.07.017 (2009).

[3] P.Yang, Y.H.Yang, B.B.Zhou, A.Y.Zomaya: A Review of Ensemble Methods in Bioinformatics, *Current Bioinformatics*, Vol.5, No.4, pp.296-308, DOI:10.2174/157489310794072508 (2010).

[4] N.Ueda: Ensemble Learning, *IPSJ Transactions on Computer Vision and Image Media*, Vol.46, No.SIG15(CVIM12), pp.11-20 (2005) (in Japanese).

[5] L.Breiman: Bagging Predictors, *Machine Learning*, Vol.24, No.2, pp.123-140 (1996).

[6] L.Breiman: Random Forests, *Machine Learning*, Vol.45, No.1, pp.5-32 (2001).

[7] A.Liaw, M.Wiener: Classification and Regression by randomForest, *The Newsletter of the R Project*, Vol.2/3, pp.18-22 (2002).

[8] T.G.Dietterich, G.Bakiri: Solving Multiclass Learning Problems via Error-Correcting Output Codes, *Journal of Artificial Intelligence Research*, Vol.2, pp.263-286 (1995).

[9] K.Nakata, T.Suzuki: Evaluating the Risk of Nonlinear Prediction with the Bagging Algorithm, *IEICE technical report, NLP2011-60, CAS2011-33*, Vol.111, No.243, pp.1-6 (2011) (in Japanese).

[10] K.Kondo, K.Kikuchi, S.Hotta, H.Shibuya, S.Maeda: Defect Classification Using Random Feature Selection and Bagging, *The Journal of the Institute of Image Electronics Engineers of Japan*, Vol.38, No.1, pp.9-15 (2009) (in Japanese).

[11] K.Takahashi: Ensemble Learning with Support Vector Machines by Using Class Membership Probabilities, *The 24th Annual Conference of the Japanese Society for Artificial Intelligence*, Paper ID:No.1A1-3 (2010) (in Japanese).

[12] Y.Freund, R.E.Schapire: A Decision-theoretic Generalization of On-line Learning and An Application to Boosting, *Journal of Computer and System Sciences*, Vol.55, No.1, pp.119-139 (1997).

[13] J.Friedman, T.Hastie, R.Tibshirani: Additive Logistic Regression: A Statistical View of Boosting, *Annals of Statistics*, Vol.28, No.2, pp.337-374 (2000).

[14] A.Torralba, K.P.Murphy, W.T.Freeman: Sharing Features: Efficient Boosting Procedures for Multiclass Object Detection, *Proceedings of IEEE Computer Society Conferenceon Computer Vision and Pattern Recognition*, pp.762-769 (2004).

[15] R.A.Jacobs, M.I.Jordan, S.J.Nowla, G.E.Hinton: Adaptive Mixtures of Local Experts, *Neural Computation*, Vol.3, pp.79–87 (1991).

[16] Y.Yasumura, K.Uehara: An Ensemble Learning Method Integrating Bagging and Boosting, *The 19th Annual Conference of the Japanese Society for Artificial Intelligence*, Paper ID:No.3F1-01 (2005) (in Japanese).

[17] I.Hayashi, S.Tsuruse: A Proposal of Boosting Algorithm for Brain-Computer Interface Using Probabilistic Data Interpolation, *IEICE Technical Report*, Vol.109, No.461, pp.303-308 (2010) (in Japanese).

[18] I.Hayashi, S.Tsuruse, J.Suzuki, R.T.Kozma: A Proposal for Applying pdi-Boosting to Brain-Computer Interfaces, *Proceedings of 2012 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE2012) in 2012 IEEE World Congress on Computational Intelligence (WCCI2012)*, pp.635-640 (2012).

[19] H.Irie, I.Hayashi, T.Katada: Vehicle Type Discrimination in Large-scale Outdoor Parking Lot Using pdi-Bagging, *The Symposium on Fuzzy, Artificial Intelligence, Neural Networks and Computational Intelligence(FAN2021)*, pp.207-212 (2021) (in Japanese).

[20] H.Irie, I.Hayashi: Performance Evaluation of pdi-Bagging by Generation of Correct - Error Virtual Data, *The 29th Symposium on Fuzzy, Artificial Intelligence, Neural Networks and Computational Intelligence(FAN2019)*, Paper ID:No.A3-3 (2019) (in Japanese).

[21] H.Irie, I.Hayashi: Proposal of Class Determination Method for Generated Virtual Data in pdi-Bagging, *The 34th Annual Conference of the Japanese Society for Artificial Intelligence*, Paper ID:No.103-GS-8-04 (2020) (in Japanese).

[22] H.Ichihashi, T.Watanabe: Learning Control by Fuzzy Models Using a Simplified Fuzzy Reasoning, *Journal of Japan Society for Fuzzy Theory and Systems*, Vol.2, No.3, pp.429-437 (1990) (in Japanese).

[23] H.Nomura, I.Hayashi, N.Wakami: A Self-Tuning Method of Fuzzy Control by Descent Method, *The 4th International Fuzzy Systems Association Congress, Engineering*, pp.155-158 (1991).

[24] H.Irie, I.Hayashi: Design Evaluation of Learning Type Fuzzy Inference Using Trapezoidal Membership Function, *Journal of Japan Society for Fuzzy Theory and Intelligent Informatics*, Vol.31, No.6, pp.908-917 (2019) (in Japanese).