pdi-Bagging: A Proposal of Bagging-Type Ensemble Method Generating Virtual Data

Honoka Irie and Isao Hayashi
Graduate School of Informatics, Kansai University, Takatsuki, Osaka, Japan

Keywords: Fuzzy Inference, Virtual Data, Ensemble Method, Bagging, Clustering.

Abstract: For pattern classification problems, there is ensemble learning method that identifies multiple weak classifiers by the learning data and combines them together to improve the discrimination rate of testing data. We have already proposed pdi-Bagging (Possibilistic Data Interpolation-Bagging) which improves the discrimination rate of testing data by adding virtually generated data to learning data. In this paper, we propose a new method to specify the generation area of virtual data and change the generation class of virtual data. As a result, the discriminant accuracy is improved since five new bagging methods which generate virtual data around correct discrimination data and error discrimination data are formulated, and the class of virtual data is determined with the proposed new evaluation index in multidimensional space. We formulate a new pdi-Bagging algorithm, and discuss the usefulness of the proposed method using numerical examples.

1 INTRODUCTION

Recently, ensemble learning methods(Polikar, 2006; Rokach, 2009), which are useful for pattern classification problems, have been proposed. The ensemble method learns multiple weak classifiers using training data and can improve the classification accuracy of the evaluation data by combining multiple weak classifiers over the layers. Ensemble learning can be broadly categorized into two types: the classifier combination model and the attribute combination model. In particular, the classifier combination model can be classified into an independent type in which each classifier is combined independently and a dependent type in which each classifier is combined while maintaining a dependency relationship. In the independent type, each classifier is trained with individual training data, so it is possible to integrate them independently and obtain a high classification rate. The independent type includes the bagging method(Breiman, 1996), Random Forests(Breiman, 2001), Error-Correcting Output Codes(Dietterich and Bakiri, 1995). The bagging method represents bootstrap aggregation. The learning data for a classifier are obtained via bootstrap sampling, and multiple classifiers are learned independently from the learning data. Finally, the final result is obtained based on the majority vote involving all the integrated classifiers. Since 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(Breiman, 1996).

On the other hand, there are boosting method(Freund and Schapire, 1997; Friedman et al., 2000) and adaptive mixture method of local experts(Jacobs et al., 1991) as the dependent type of classifier combination model. Boosting is a method to improve the classification rate by sequentially learning weak classifiers. In particular, AdaBoost(Freund and Schapire, 1997) is particularly useful and has the advantage of being easy to analyze features of datasets. In this way, the dependent type, represented by boosting, is trained by multiple weak classifiers while maintaining sequential interdependence with the training data and can identify the input-output relationship with the dependence. On the other hand, in the independent type, represented by bagging, the weak classifier is independent for each training data, but the processing algorithm is relatively simple and has high accuracy.

We have proposed a new bagging algorithm for the generation and interpolation of data around misclassified data using a specified membership function(Hayashi and Tsuruse, 2010; Hayashi et al., 2012). We name this method possibilistic data interpolation bagging (pdi-bagging). The interpolation of data around misclassified data is called vir-

Irie, H. and Hayashi, I.
pdi-Bagging: A Proposal of Bagging-Type Ensemble Method Generating Virtual Data.
DOI: 10.5220/0011827600003393
In *Proceedings of the 15th International Conference on Agents and Artificial Intelligence (ICAART 2023) - Volume 3*, pages 956-963
ISBN: 978-989-758-623-1; ISSN: 2184-433X

tual data. In pdi-bagging, data misclassified by the classifier model are not weighted as in AdaBoost, nor are they added to the next training data. The classes of virtual data are estimated from their locations(Irie and Hayashi, 2019b; Irie and Hayashi, 2020) and the virtual data are added to training data to estimate discriminant lines using weak classifiers based on fuzzy inference(Nomura et al., 1991; Irie and Hayashi, 2019a). 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 finally, the classification rate for the evaluation data is obtained by a majority vote of multiple weak classifiers. Since the number 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 identifying the discriminant line. In this paper, we formulate five types of virtual data generation methods and discuss their usefulness using numerical examples.

2 pdi-Bagging

A conceptual diagram of pdi-bagging is shown in Fig.1. In pdi-Bagging, first, weak classifiers M_0 of fuzzy inference are learned using training data probabilistically extracted from all datasets, and the discriminant rate of the training data TRD is calculated. Next, virtual data are generated around the misclassified data using membership functions. The generated virtual data is added to the original training data to increase the number of training data TRD. Using original training data and virtual data, the classification rate is calculated by a weak classifier M_1 based on fuzzy inference. Because of increasing the number of TRD improves the discriminant accuracy of weak classifiers. The repeating of operations is finished at the L times when the end judgment is satisfied. Finally, the evaluation data (CHD) are input to L weak classifiers $M_0, M_1, \cdots, M_l, \cdots, M_L$, and the final result is then calculated by majority rule. Since pdi-Bagging adds virtual data to training data and calculates the discriminant rate by multiple weak classifiers, its discriminant rate is higher than the conventional bagging method and AdaBoost(Hayashi and Tsuruse, 2010; Hayashi et al., 2012).

In pdi-Bagging, fuzzy clustering by simple fuzzy inference(Nomura et al., 1991) is adopted as a weak classifier. Fuzzy inference is excellent in learning ability and can realize visualization of learning results using rule description. Therefore, the fuzzy inference is adopted here as a weak classifier. Simplified fuzzy inference expresses rules in if-then form, uses fuzzy sets defined by membership functions in the antecedent part, and defines the consequent part in singleton form with real numbers. We use here a trapezoidal fuzzy set as the membership function.

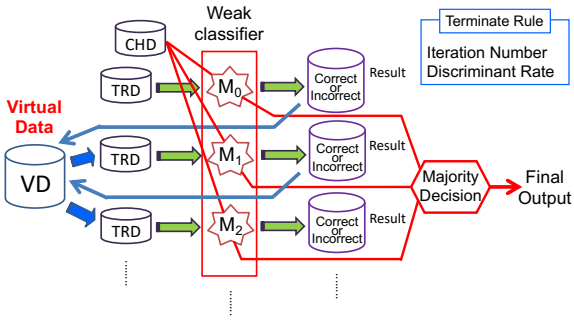


Figure 1: pdi-Bagging Algorithm.

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

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

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

Suppose we have obtained the input data $\mathbf{x} = (x_1, x_2, \cdots, x_n)$. The input data $\mathbf{x}$ is input to 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 result of fuzzy inference, $\hat{z}$, and class C are calculated by the following equations.

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

Now, let's explain how to generate virtual data in 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 data set D consisting of W data. 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, $0 \leq h \leq 1$, 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.

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

where, $x_j^S(d)$ means correct-classified data $x_j^C(d)$ or misclassified data $x_j^E(d)$. In addition, the membership

function $\mu_F(x_j)$ is defined by 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 by the 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 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.

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

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

In particular as to CC, we explain about how to generate virtual data arround the correct classified data at the cluster center using Fig.2. We assume the classtering problem of a total of 8 data into two classes, green class and yellow class, in Fig.2. The training data with a green frame in yellow located at the bottom of the figure is misclassified as the green class, although the true class is the yellow class. Since the true class of the misclassified data $\mathbf{x}^E(d)$ is yellow class, the midpoint between the closest correct classified data and the farthest correct classified data from $\mathbf{x}^E(d)$, whose classes are yellow is calculated. Virtual data $\mathbf{x}^V(d')$ are generated around the correct classified data $\mathbf{x}^C(d')$ whose class is yellow closest to the midpoint. According to the generation method, many virtual data in the CC method tend to generate 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, compared to the CA that generates virtual data in the entire space. In addition, it is possible to control the degree of influence on the discriminant line by moving the coordinate posision currently set as the midpoint to an arbitrary interpolation point or extrapolation point from the endpoints.

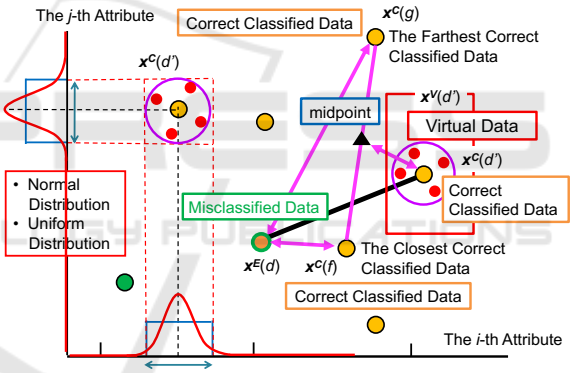


Figure 2: Generation of Virtual Data with Correct Data around Cluster Center.

3 FORMULATION FOR CLASS MODIFICATION

We propose here a new class determination method for assigning correct classes to 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 $\mathbf{x}^E(d)$. Basically, 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 generate at locations far from the source data. In addition, virtual data may generate in areas where different classes of data are dense. herefore, the class k^* of the virtual data $\mathbf{x}^V(d)$ is determined by the inte-

gration 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 by the distance between the virtual data $\mathbf{x}^V(d)$ and the source data $\mathbf{x}^{S,k}(d)$ with class k. The smaller this evaluation value E_1, the higher the dependence of $\mathbf{x}^V(d)$ on 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 by the distance between the virtual data $\mathbf{x}^V(d)$ and the center of the class k. The smaller this evaluation value E_2, the higher the dependence of $\mathbf{x}^V(d)$ on class k. Now, when the center of class k is represented by $\mathbf{x}_c^k$,

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

(3)Evaluation of Neighborhood Data Classes(E_3)

The evaluation value E_3 is defined by the distance between the virtual data $\mathbf{x}^V(d)$ and the closest correct/misclassified data $\mathbf{x}^{S,k}(e)$ with class k. The smaller this evaluation value E_3, the higher the dependence of $\mathbf{x}^V(d)$ on class k.

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

According to these three criteria, the evaluation E_1 is higher when the virtual data generate near the source data. On the other hand, 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 is obtained. The virtual data $\mathbf{x}^V(d)$ has the class k^* that minimizes the following overall evaluation value E^k.

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

where w_1, w_2, w_3 are the weights of each evaluation value.

We formulate the pdi-Bagging algorithm as follows.

Step 1 We assume that the W data D is obtained. Data D are categorized into two types of datasets: W^{TRD} training data D^{TRD} and W^{CHD} check data D^{CHD}. In addition, interpolated data are represented by D^V.

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

Step 3 The d-th data that was correctly or misclassified is temporarily extracted from D^{TRD}. Assume 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 $\mathbf{x}^V(d)$ by the equation (2). Remove the virtual data $\mathbf{x}^V(d)$ from the $l-1$th D^V with $l > 2$, and adds 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 by random number and add them to D^{TRD}.

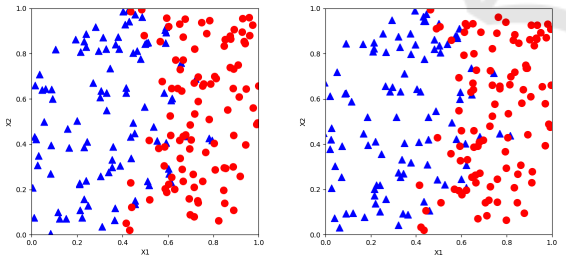
Step 6 Steps 2 to 4 are repeated with $l = l + 1$, and the algorithm is terminated at $K = l$ satisfying $r_l^{CHD} \geq \theta$ for threshold θ. Alternatively, the algorithm ends when $l \geq K$ is satisfied for the number of weak classifiers L and the number of iterations K, $K \leq L$.

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.

4 VERIFICATION AND DISCUSSION USING NUMERICAL DATA

To explain the pdi-bagging algorithm, we discuss 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 for training data 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-classes discrimination problems as numerical data. For this discriminant problem, the real value of the consequent part of the fuzzy inference rules are set to 2.0 (red, ◯) and 3.0 (blue, △).

Simplified fuzzy inference is used as the weak classifier, and five types of trapezoidal membership functions are set for each input interval [0, 1]. Since the data space is two-dimensional, the 25 rules are constracted in the whole area of the space. In addition, in order to verify the classification rate when rules are added to the data space as specific areas, 49 rules are 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 4 rules are added to $G_2 = \{(x_1, x_2) \mid [0.7, 0.8] \times [0.3, 0.7]\}$ as the specific area G_2. As a result, the total number of rules is 78. The addition of the rules improves the accuracy of the discriminant rate in regions away from the discriminant line where the data are dense, and the overall discriminant rate is improved. The discriminant rate was here calculated for a total of 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 the specific region are set as right-angled trapezoid type at both ends of specific regions, the size of the specific region does not change even if the membership functions are learned. On the other hand, when trapezoidal membership functions are set at both ends of specific regions, the size of the region changes as the membership functions are learned. Therefore, when the right-angled trapezoidal membership function are set in the additional rules, the membership functions do not move outside the specific region even when the membership functions are learned, and it is learned intensively within the specific region.



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

Figure 3: Numerical Example Training and Testing Data.

The initial value of the antecedent part of the fuzzy reasoning is set by the default method, and the learning order of the antecedent and consequent parts is that the consequent part is learned first, and then the antecedent part and the consequent part are alternately learned. In 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 denote K_b and K_c, and were set to 0.01(Irie and Hayashi, 2019a). 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(Irie and Hayashi, 2019a). On the other hand, 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 is set to 10, and the alternating learning of the consequent part is set to (10, 10).

As a membership function $\mu_F(x_j)$ for generating virtual data, the normal distribution of Equation (1) with a standard deviation of σ = 0.5 was selected, andthe number of virtual data generated was basically one. However, in preliminary experiments, the discriminant rate of fuzzy inference was about 87%. As a result, about 26 out of 200 checking data are erroneously classified, and about 8 virtual data are required to make the total number of virtual data equal to 200 training data. Therefore, we also discussed the discriminant rate when the number of generated virtual data was changed from 1 to 10.

The evaluation values weight for class estimation of virtual data are $(w_1, w_2, w_3) = \{$(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), (0.05, 0.475, 0.475)$\}$. In determining the weight, the weight w_1 of the distance from the source data has a large effect on the class estimation. Therefore, we discussed the discriminant rate for a total of 7 types: $w_1 = 1/3$ when $w_1 = w_2 = w_3$, $w_1 = 0.5$, and 5 types with the value of w_1 reduced.

The algorithm is terminated by the termination rule whose number of iterations $K = 5$. 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 learning process of fuzzy inference, the order of data is changed by random numbers every epoch. Since the number of epochs for the learning of the consequent part and the alternate learning of the antecedent part and the consequent part is 10 and (10, 10), respectively, the total number of epochs is 150 in the five-layer learning. Since 2-fold cross-validation is used here, 150 epochs of epoch learning for each data set to result in a total of 300 epochs of learning. We compared the average discriminant rates obtained in 10 trials for each of the different types, CA, CC, E, MA, and MC.

The discriminant rate for evaluation data by 5 types of virtual data generation methods: type of correct classified data in the whole space(CA), type of correct classified data at the cluster center(CC), type of misclassified data(E), mixing type of correct classified

data and misclassified data in the whole space(MA), and mixing type of correct classified data and misclassified data at the cluster center(MC) are shown in Table 1 and Figures 4-6. Table 1 shows the discriminant rate for each weight with respect to the evaluation index, with and without additional rules, and with respect to the shape of the membership function within a specific region. We also calculated the difference from the discriminant rate when 25 rules were set with the trapezoidal membership function. Figures 4 to 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 shape of the membership function within a specific region.

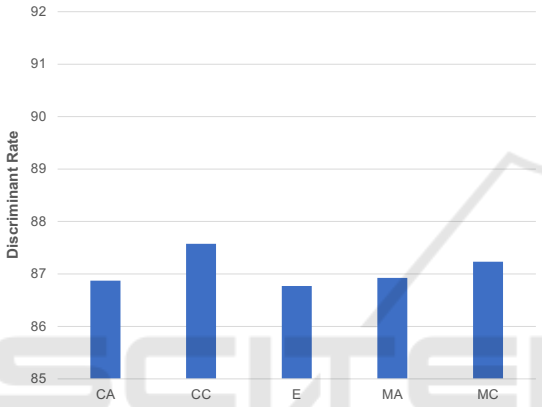


Figure 4: Average Discriminant Rates of 5 Methods in 25 Basic Rules.

First, from the results in Table 1 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 2-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, so the generation of virtual data is effective in improving the discriminant rate.

In the case of 25 rules in the trapezoidal membership function, the discriminant rate is not necessarily high. On the other hand, the discriminant rates of 5 methods are higher than that of the 25 rules. In the types of correct classified data, the discriminant rate of CC is higher than that of CA, and even in the mixing types of correct classified data, the discriminant rate of MC is higher than that of MA. The reason is that in CC and MC, the virtual data are generated near the center of the cluster, so the fuzzy rules near the center of the class are learned with high accuracy.

Table 1 and Fig.5 show the characteristics of the discriminant rate of the 78 rules added within the specific region by the trapezoidal membership function. The discriminant rate of 2-fold cross validation of simple fuzzy inference with 78 rules using the trapezoidal membership function was 89.68%. On the other hand, among the five types of virtual data generation methods, the discriminant rates of three types, CC, E, and MC are higher than simple fuzzy inference. Therefore, methods other than generating virtual data in the entire space are effective.

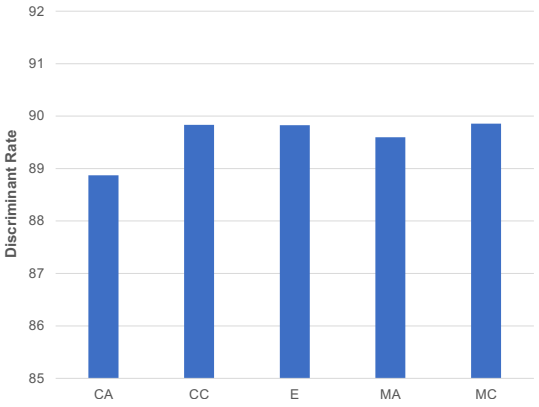


Figure 5: Average Discrimination Rates of 5 Methods in 78 Total Rules Added by Trapezoidal Membership Function.

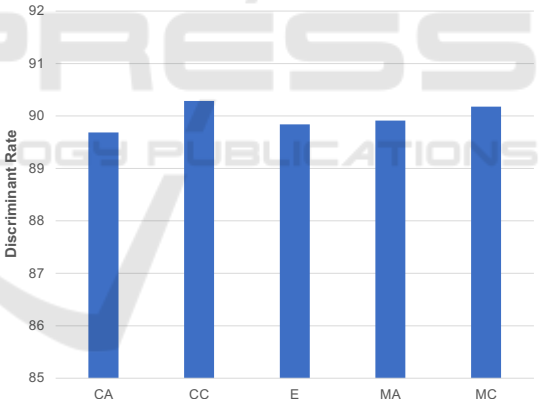


Figure 6: Average Discrimination Rates of 5 Methods in 78 Total Rules Added by Right Trapezoidal Membership Function.

In addition, Table 1 and Fig.6 show the characteristics of the discriminant rate of the 78 rules added within the specific region by the right-angled trapezoidal membership function. The discriminant rate of 2-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 four types, CC, E, MA, and MC are higher than simple fuzzy inference. In particular, the MC and the CC are higher than 0.45%. Therefore, in the case of 78 rules with right-angled trapezoidal membership

Table 1: Comparison of Discriminante Rates According to 5 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

functions, the average discriminant rate is high for CC and MC. From 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 is slightly lower than the other methods. In addition, the average discriminant rate of the 78 rules of the right-angled trapezoidal membership function is 0.38% higher than that of the 78 rules of the trapezoidal membership function. On the other hand, the maximum discriminant rate was 90.35% for CC when the weight of the evaluation index was (0.2, 0.5, 0.3) and MC when the weight of the evaluation index was (0.5, 0.25, 0.25). In the specific area, there are a lot of singular point data, so the learning of the rules in this area increases the overall discriminant rate. In addition, when the right-angled trapezoidal membership functions are set in this specific region, the size of the specific region does not change, so the membership functions are efficiently learned within the specific region, and the overall discriminant rate increases.

Table 2: Results of t-Test between 5 Methods in 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	—

Table 2 shows the results of the t-test of the discriminant rate by five virtual data generation methods using 25 rules of the trapezoidal membership function. The numerical data in Fig.3 were used alternately as training data and checking data by 2-fold cross validation. In Table 2, the significance of each data is indicated by ① and ② when there is a significant difference between the five methods in the one-tailed t-test with a significance level of 5%. In addition, the average value of p is shown when only one of ① and ② is significant. From Table 1, the discriminant rates of CA, E, and MA are low, and the discriminant rates of CC and MC are high. Therefore, CC and MC are useful methods with higher discriminant rate than other methods.

Summarizing the results, the methods with the highest discriminant rate were CC and MC with 78 rules using right-angled trapezoidal membership functions in specific regions. In the two methods, the discriminant rate was improved by adding rules to specific regions where singularity data exists. In addition, since 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 have led to high discriminant rates.

5 CONCLUSIONS

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

In the future, it is necessary to discuss how to generate virtual data when there is a bias in the amount of data between classes, and how to generate virtual data with directionality. In addition, it is necessary to discuss the usefulness of pdi-Bagging in practical applications using actual measurement data.

ACKNOWLEDGEMENTS

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

REFERENCES

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2):123–140.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Dietterich, T. G. and Bakiri, G. (1995). Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2:263–286.

Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139.

Friedman, J., Hastie, T., and Tibshirani, R. (2000). Additive logistic regression: A statistical view of boosting. *Annals of Statistics*, 28(2):337–374.

Hayashi, I. and Tsuruse, S. (2010). A proposal of boosting algorithm for brain-computer interface using probabilistic data interpolation. *IEICE Technical Report*, 109(461):303–308 (in Japanese).

Hayashi, I., Tsuruse, S., Suzuki, J., and Kozma, R. T. (2012). A proposal for applying pdi-boosting to brain-computer interfaces. In *Proceedings of 2012 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE2012) in 2012 IEEE World Congress on Computational Intelligence (WCCI2012)*, pages 635–640.

Irie, H. and Hayashi, I. (2019a). Design evaluation of learning type fuzzy inference using trapezoidal membership function. *Journal of Japan Society for Fuzzy Theory and Intelligent Informatics*, 31(6):908–917 (in Japanese).

Irie, H. and Hayashi, I. (2019b). Performance evaluation of pdi-bagging by generation of correct - error virtual data. In *The 29th Symposium on Fuzzy, Artificial Intelligence, Neural Networks and Computational Intelligence(FAN2019)*, pages Paper ID:No.A3–3 (in Japanese).

Irie, H. and Hayashi, I. (2020). Proposal of class determination method for generated virtual data in pdi-bagging. In *The 34th Annual Conference of the Japanese Society for Artificial Intelligence*, pages Paper ID:No.103–GS–8–04 (in Japanese).

Jacobs, R. A., Jordan, M. I., Nowla, S. J., and Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural Computation*, 3:79—-87.

Nomura, H., Hayashi, I., and Wakami, N. (1991). A self-tuning method of fuzzy control by descent method. In *The 4th International Fuzzy Systems Association Congress, Engineering*, pages 155–158.

Polikar, R. (2006). Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine*, 6(3):21–45.

Rokach, L. (2009). Taxonomy for characterizing ensemble methods in classification tasks: A review and annotated ibliography. *Computational Statistics & Data Analysis*, 53(12):4046–4072.