

正誤バーチャルデータの発生による pdi-Bagging の特性評価

Performance Evaluation of pdi-Bagging by Generation of Correct - Error Virtual Data

○ 入江 穂乃香
○ Honoka Irie
関西大学大学院
Kansai University

林 勲
Isao Hayashi
関西大学大学院
Kansai University

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 observation data. In this paper, we formulate three new bagging methods which generate virtual data around correct discrimination data and error discrimination data, and discuss the characteristics of the bagging methods by the numerical examples.

1. はじめに

機器によって計測された観測データを分析する場合、観測データを各状態クラスに分類して、データクラスの識別線を推定することが行われる。このデータクラスの識別線推定のためにクラスタリング手法がよく用いられる。一般的に、クラスタリングの推定精度を向上させるには、十分な量の学習データが必要である。しかし、実際の計測現場では、クラス間でデータ量の偏りが生じる場合が多い。具体的には、正常に分類された計測データは多いが、異常として分類された計測データは少ない。このようなクラス間の偏りが生じた場合、従来の方法では、正常クラスと異常クラスにデータ量の偏りが無いように正常データの量を調整し、クラスタリング法によって識別線を推定する方策をとっていた。

一方、パターン分類問題に対して、複数個の弱識別器を同定し、それらを統合的に組み合わせることで全体の識別精度を向上させるアンサンブル学習 [1-4] に対する関心が高まっている。アンサンブル学習の一手法にバギング法 (Bagging Methods) がある。バギング法とは、データ集合からサンプリングによって複数の学習データを構成し、それらの学習データを用いて複数個の弱識別器を同定して、評価データに対する高い識別率を得る手法である。ブースティング法が繰り返し学習の際に、複数の学習データ間で相互依存関係を有するのに対して、バギング法では、複数の学習データ間は独立である。我々は、識別器の同定の際に、仮想的に生成したデータを観測データに追加して学習データを構成し、識別線の精度を向上させる新たなバギング法やブースティング法を提案している。このバギング法を pdi-Bagging (Possibilistic Data Interpolation-Bagging) と呼び、仮想的に生成したデータをバーチャルデータと呼ぶ [5-8]。バーチャルデータの追加により、学習時にデータ量が増えるので、クラス

間のデータ量の偏りがなくなり、識別線の同定精度が向上する。従来の pdi-Bagging では、バーチャルデータは誤識別データ周辺に生成する。しかし、バーチャルデータは、誤識別周辺だけでなく正識別データ周辺に生成することも可能である。

本論文では、バーチャルデータの生成位置による pdi-Bagging の識別精度と特性について議論する。ここでは、バーチャルデータはメンバシップ関数を用いて発生する。発生メカニズムとして、学習データの識別結果が正識別であるデータの周辺にバーチャルデータを発生させるアルゴリズムや誤識別のデータの周辺にバーチャルデータを発生させるアルゴリズムを定式化する。誤識別クラスのデータは識別線に近い位置に存在するので、誤識別データ周辺で発生したバーチャルデータは識別線近傍に分布する。一方、正識別クラスのデータは識別線の近傍に位置するとは限らず、発生したバーチャルデータは全データ空間に均一に分布する。また、正識別と誤識別の混合型のアルゴリズムも定式化する。これらのバーチャルデータは観測データ集合に追加され学習データを構成する。pdi-Bagging では、バーチャルデータを追加した学習データをファジィ推論による弱識別器に入力し識別線を推定して、次層でも同様にバーチャルデータの追加による学習データの構成と識別線の推定を行う。これらの一連の操作を繰り返し、最終的には、弱複数個の識別器の多数決によって評価データの識別率を得る。したがって、バーチャルデータの発生位置は重要な意味をもつ。ここでは、これらのバーチャルデータの発生位置の違いによる pdi-Bagging の識別精度と特性を数値例により議論する。

従来のバギング法が、全データ集合からサンプリングによってデータ数を減じて学習データを構成するのに対して、pdi-Bagging は、バーチャルデータの追加によってデータ数を増加して学習データを構成する。したがって

て、pdi-Bagging では、多量の学習データ集合によって識別線が同定されるので、従来のバギング法と比較して識別線がより滑らかで評価データの識別率が良い。また、バーチャルデータを発生するメカニズムを有しているため、観測データ数の不足やデータの偏りによる識別器の信頼度低下も防ぐことができる。

2. pdi-Bagging の定式化

バギング法とは、複数個の識別器を用意し、各識別結果を統合することにより評価データに対する高い識別率を得る手法である。ブーティング法が繰り返し学習の際に、複数の学習データ間で相互依存関係を有するのに対して、バギング法では、複数の学習データ間は独立である。さらに、pdi-Bagging では、バーチャルデータの追加により学習データ量は増加する。

一般的なバギング法では、まず、全データ集合から確率的に抽出された学習データ (TRD) を用いて識別器 M_1 を学習し、 TRD の識別率を算出する。次ステップ (層) で、全データ集合から新たに確率的に学習データ TRD を抽出して構成し、識別器 M_2 により TRD の識別率を算出する。終了判定が満足されるまでこの一連の操作を L 回繰り返す。最終的に、評価データ (CHD) を L 個の識別器 $M_1, M_2, \dots, M_i, \dots, M_L$ に入力して、多数決により最終結果を得る。

一方、pdi-Bagging では、メンバシップ関数を用いてバーチャルデータを特定の観測データの近傍に発生させて TRD を増加させる。 TRD を増加させることによって識別器の識別精度が向上する。次層でも、バーチャルデータを特定の観測データの近傍に発生させて新たな TRD を構成し、識別器 M_2 により TRD の識別率を算出する。終了判定が満足されるまでこの一連の操作を L 回繰り返す。最終的に、評価データ (CHD) を L 個の識別器 $M_1, M_2, \dots, M_i, \dots, M_L$ に入力して、多数決により最終結果を得る。pdi-Bagging の概念図を図1に示す。また、アンサンブル学習で一般に多用されている AdaBoost と pdi-Bagging の違いを図2に示す。pdi-Bagging では、バーチャルデータが学習データに追加されて識別器が同定されるので、従来のバギング法や AdaBoost よりも高い精度の識別線が構成される [6]。

pdi-Bagging では、弱識別器として簡易型ファジィ推論 [9] によるファジィクラスタリングを用いる。簡易型ファジィ推論は、if-then 型のルールをもち、前件部では、メンバシップ関数のファジィ集合を定義し、後件部では、シングルトンを定義する。ここでは、三角型のメンバシップ関数を一般化した正規な台形型ファジィ集合を用いる。

いま、入力変数 x_j , $j = 1, 2, \dots, n$ の台形型ファジィ集合 $\mu_F(x_j)$ を次の $L - R$ ファジィ集合で定義する。

$$\mu_F(x_j) = \begin{cases} L(\frac{x_b - x_j}{\alpha}) & ; x_j \leq x_b \\ 1.0 & ; x_b \leq x_j \leq x_c \\ R(\frac{x_j - x_c}{\beta}) & ; x_j \geq x_c \end{cases} \quad (1)$$

ただし、下底の左端、上底の左端、上底の右端、下底の右端の座標をそれぞれ、 (x_a, y_a) , (x_b, y_b) , (x_c, y_c) , (x_d, y_d) で表し、左の広がり $\alpha = |x_b - x_a|$ で、右の広がり $\beta = |x_d - x_c|$ で表す。なお、 $L()$, $R()$ は型

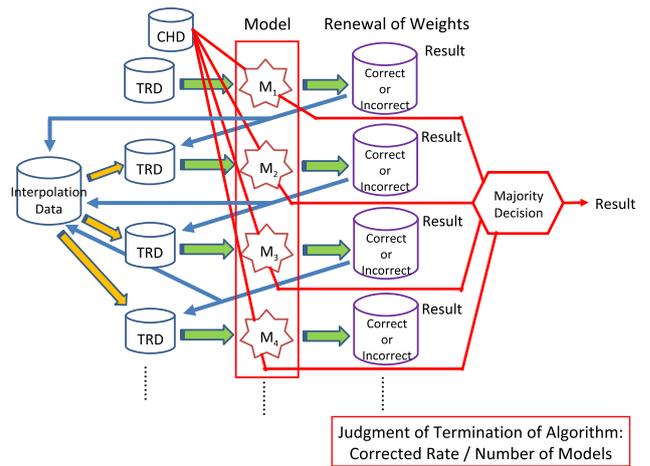


Fig. 1: pdi-Bagging Algorithm

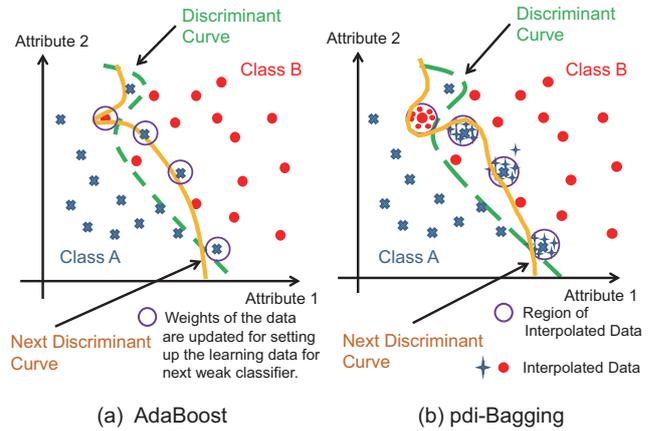


Fig. 2: AdaBoost and pdi-Bagging

関数と呼ばれ、 $\max(0, 1 - |x|)$ なる関数である。

いま、出力変数を z 、後件部のシングルトンを p_i で表すと、ファジィルール r_i , $i = 1, 2, \dots, R$ は次のようになる。

$$\begin{aligned} r_1 : & \text{ if } x_1 \text{ is } \mu_{F_{11}}(x_1) \text{ and } \dots \text{ and } x_n \text{ is } \mu_{F_{1n}}(x_n) \\ & \text{ then } C = \{C_{1k} \mid z = p_1\} \\ & \vdots \\ r_i : & \text{ if } x_1 \text{ is } \mu_{F_{i1}}(x_1) \text{ and } \dots \text{ and } x_n \text{ is } \mu_{F_{in}}(x_n) \\ & \text{ then } C = \{C_{ik} \mid z = p_i\} \\ & \vdots \\ r_R : & \text{ if } x_1 \text{ is } \mu_{F_{R1}}(x_1) \text{ and } \dots \text{ and } x_n \text{ is } \mu_{F_{Rn}}(x_n) \\ & \text{ then } C = \{C_{Rk} \mid z = p_R\} \end{aligned}$$

ただし、 C は出力クラスの変数であり、 C_{ik} はルール r_i のクラス値が C_k であることを示す。

いま、入力データ $\mathbf{x} = (x_1, x_2, \dots, x_n)$ が得られたとしよう。第 i 番目のファジィルール r_i の前件部に入力

データ \mathbf{x} を入力し, 前件部の適合度 $\mu_i(\mathbf{x})$ を計算する.

$$\mu_i(\mathbf{x}) = \mu_{F_{i1}}(x_1) \cdot \mu_{F_{i2}}(x_2) \cdot \cdots \cdot \mu_{F_{in}}(x_n) \quad (2)$$

ファジィ推論の結果 \hat{z} とクラス C は次式から求める.

$$\hat{z} = \frac{\sum_{i=1}^R \mu_i(\mathbf{x}) \cdot p_i}{\sum_{i=1}^R \mu_i(\mathbf{x})}$$

$$C = \{C_k \mid \min |\hat{z} - z|\}$$

さて, pdi-Bagging におけるバーチャルデータの生成方法について説明しよう. W 個のデータからなるデータ集合 D において, 第 d 番目のデータを $\mathbf{x}(d) = (x_1(d), x_2(d), \dots, x_n(d))$ で表す. ある特定のデータ $\mathbf{x}^S(d)$ の周辺にバーチャルデータ $x_j^V(d)$ を発生する. バーチャルデータの発生方法として次の 3 種類を提案する.

正識別型: 弱識別器により $\mathbf{x}^S(d)$ のクラスが正しく識別 (正識別) された場合に, その周辺にバーチャルデータ $x_j^V(d)$ を発生させる.

誤識別型: 弱識別器により $\mathbf{x}^S(d)$ のクラスが誤って識別 (誤識別) された場合に, その周辺にバーチャルデータ $x_j^V(d)$ を発生させる.

混合型: バギングの各層ごとに, 正識別型と誤識別型の交互として, $\mathbf{x}^S(d)$ の周辺にバーチャルデータ $x_j^V(d)$ を発生させる.

ここでは, これらの 3 種類のバーチャルデータの発生方法の特性を議論することが目的である.

バーチャルデータ $x_j^V(d)$ の発生位置は, ある実数 h , $0 \leq h \leq 1$ が与えられると, ファジィ数 F のメンバシップ関数 $\mu_F(x_j)$ を用いて, 各次元ごとに次のように発生する.

$$x_j^V(d) = \{x_j \mid \mu_F(x_j) = h, \mu_F(x_j^S(d)) = 1\} \quad (3)$$

$$h \sim N(1, 1), \quad 0 \leq h \leq 1 \quad (4)$$

ただし, ファジィ数 F の中心は $x_j^S(d)$ である.

メンバシップ関数 $\mu_F(x_j)$ には, 次のような L 関数や正規分布, 一様分布などがある.

$$\mu_F(x_j) = L\left(\frac{x_j - x_j^S(d)}{c}\right), \quad c \geq 0 \quad (5)$$

$$\mu_F(x_j) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x_j - x_j^S(d))^2}{2\sigma^2}\right) \quad (6)$$

$$\mu_F(x_j) = \begin{cases} \frac{1}{x_j^{max} - x_j^{min}} & ; x_j^{min} \leq x_j \leq x_j^{max} \\ 0 & ; x_j < x_j^{min}, x_j > x_j^{max} \end{cases} \quad (7)$$

ただし, c はメンバシップ関数の幅であり, $L(x)$ 関数の例として $L(x) = \max(0, 1 - |x|)$ の三角型メンバシップ関数などがある. また, σ は正規分布の標準偏差であ

り, 一様分布の x_j^{max} と x_j^{min} は次の値である.

$$x_j^{min} = \frac{3x_j^S(d) + x_j^S(d_{min})}{4} \quad (8)$$

$$x_j^{max} = \frac{3x_j^S(d) + x_j^S(d_{max})}{4} \quad (9)$$

$$x_j^S(d_{min}) = \arg \min_d (x_j^S(d) - x_j(d)), \\ x_j^S(d) - x_j(d) \geq 0 \quad (10)$$

$$x_j^S(d_{max}) = \arg \min_d (x_j(d) - x_j^S(d)), \\ x_j(d) - x_j^S(d) \geq 0 \quad (11)$$

ここで, x_j^{min} と x_j^{max} はそれぞれ $x_j^S(d)$ から隣接する x_j への内分点であるが, 3:1 の比率は経験値によるものである.

pdi-Bagging のアルゴリズムを次のように定式化する.

Step 1 計測データ D (個数: W 個) を学習データ D^{TRD} (個数: W^{TRD} 個) と評価データ D^{CHD} (個数: W^{CHD} 個) に分割する. また, D から構成されるバーチャルデータを D^V で表す.

Step 2 第 i 番目の識別器 M_i に D^{TRD} を入力し, 第 i 番目の結果 R_i の識別率 r_i^{TRD} を得る.

Step 3 正識別あるいは誤識別された第 d 番目のデータを D^{TRD} から一時的に抽出する. 正識別あるいは誤識別データ $\mathbf{x}^S(d)$ の第 j 番目の属性値 $x_j^S(d)$ に対して, メンバシップ関数 $\mu_F(x_j)$ によりバーチャルデータ $x_j^V(d)$ を発生させる.

Step 4 バーチャルデータ $\mathbf{x}^V(d)$ のクラスを推定する. 推定には, バーチャルデータ $x_j^V(d)$ に対して, 正誤識別データ $x_j^S(d)$ からの評価を E_{j1} , 識別クラス中心からの評価を E_{j2} , 近傍クラスの評価を E_{j3} として, クラス k の評価値 E^k が最小となるバーチャルデータ $\mathbf{x}^V(d)$ のクラス k^* を決定する.

$$k^* = \{k \mid \min_k E^k = \min_k \sum_{j=1}^n E_{j3}^k\} \quad (12)$$

$$E_j^k = w_1 E_{j1}^k + w_2 E_{j2}^k + w_3 E_{j3}^k \quad (13)$$

ただし, w_1, w_2, w_3 は各評価値の重みである. $\mathbf{x}^V(d)$ のクラスは, 評価値 E^k が最小となるクラス k^* とする. バーチャルデータ $\mathbf{x}^V(d)$ を D^V に追加する.

Step 5 結果 R_i での正識別データ数と誤識別データ数が同数になるように, 乱数により D^V から $v \geq \frac{W}{2} - W^{TRD}(1 - r_i^{TRD})$ 個のバーチャルデータを取り出し D^{TRD} に加える.

Step 6 $i = i+1$ として Step2 から 5 までを繰り返し, しきい値 θ に対して $r_i^{CHD} \geq \theta$ を満足した $K = i$ の時点, あるいは, 識別器の個数 L と繰り返し回数 K , $K \leq L$ に対して $i \geq K$ を満足した時点でアルゴリズムを終了する.

Step 7 $M_1, M_2, \dots, M_i, \dots, M_K$ に D^{CHD} を適用し、多数決により結果の識別率 r_K^{CHD} を得る。

3. 数値データによる検証と考察

正誤バーチャルデータ発生を検証に用いる数値データは、学習データとして100個、評価データとして100個の合計200個である。学習データを図3に示す。2入力1出力の2群判別問題として、乱数により各入力値を $[0, 1]$ 内で発生し、2群クラスの実数値を2.0 (赤・○印) と3.0 (青・△印) に設定した。

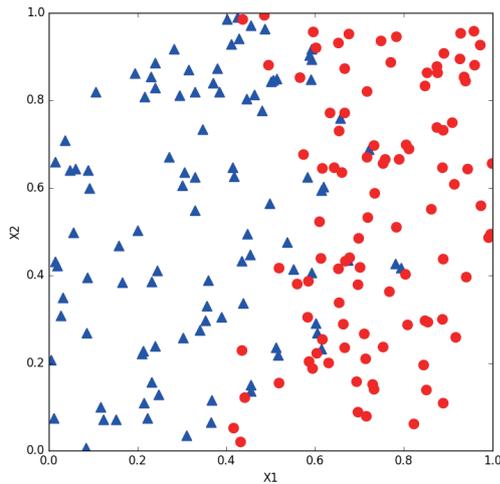


Fig. 3: Numerical Example Data

弱識別器には簡易型ファジィ推論を用いるので、各入力区間 $[0, 1]$ を5領域に分割し、5種類のメンバシップ関数を設定した。ルール数は25個であるので、学習パラメータの総数は225個となる。前件部の初期値設定は最大法とし、前件部と後件部の学習順序は、後件部→前件部交互学習とした。各入力の5種類のメンバシップ関数の学習係数 K_b, K_a, K_c, K_β は同一とし0.01に設定した。また、後件部のシングルトンの学習係数 K_p は、最初の後件部学習では0.4に設定し、交互学習の後件部学習では0.6とした。バーチャルデータ発生メンバシップ関数 $\mu_F(x_j)$ は正規分布とし、バーチャルデータのクラス推定のための評価値の重みを $w_1 = w_2 = w_3 = 1/3$ とする。アルゴリズムの終了規範は繰り返し判定として、回数は $K = 3$ とした。

データ検証の実験では、後件部と前後件部交互学習のエポック回数をそれぞれ (10, 10, 10), (10, 20, 20), (10, 30, 30) の3種類に設定する。例えば、(10, 30, 30) では、後件部が10エポック回の学習の後、前件部と後件部が交互にそれぞれ30エポック回数で学習する。1エポックごとに乱数によりデータの並び順序を変更し、これらを1試行として、10試行の正識別型、誤識別型、混合型の識別率を比較した。ただし、混合型は、第1層を誤識別型とし、第2層を正識別型とする。

エポック回数が (10, 30, 30) の混合型の場合を例にとり、pdi-Baggingのアルゴリズムを説明しよう。データ集

合 D は $W = 200$ であるので、 $W^{TRD} = W^{CHD} = 100$ である。 D_1^{TRD} を第1層目のファジィ推論の弱識別器 M_1 に入力し、結果 R_1 の識別率 $r_1^{TRD} = 86.0\%$ を得る。 R_1 における誤識別クラスをもつデータ数は28個であった。式(3)と式(4)、及び、式(6)によって、全ての誤識別データ $\mathbf{x}^S(d)$ の周辺に各1個のバーチャルデータ $\mathbf{x}^V(d)$ を発生させ D^V に保存する。誤識別データとバーチャルデータの合計個数が正識別データの個数と等しくなるまで、 D^V から誤識別のバーチャルデータを学習データに加える。 R_1 におけるバーチャルデータ数 v_1 は28個であり、第2層目の新たな学習データ集合 D_2^{TRD} は $W_2^{TRD} = 228$ となった。 D_2^{TRD} を第2層目のファジィ推論の弱識別器 M_2 に入力し、結果 R_2 の識別率 $r_2^{TRD} = 88.5\%$ を得る。177個の正識別データ $\mathbf{x}^S(d)$ の周辺に各1個のバーチャルデータ $\mathbf{x}_j^V(d)$ を発生させる。バーチャルデータ数は $v_2 = 177$ 個となり、 D_3^{TRD} は $W_3^{TRD} = 377$ となる。 R_3 では、弱識別器 M_3 による識別率が $r_3^{TRD} = 87.0\%$ となり、終了判定 $K = 3$ によって学習が終了する。多数決原理により評価データに対するファジィ推論の識別器の結果 R_1, R_2, R_3 を統合し、最終結果の識別率として $r^{CHD} = 86.0\%$ を得る。

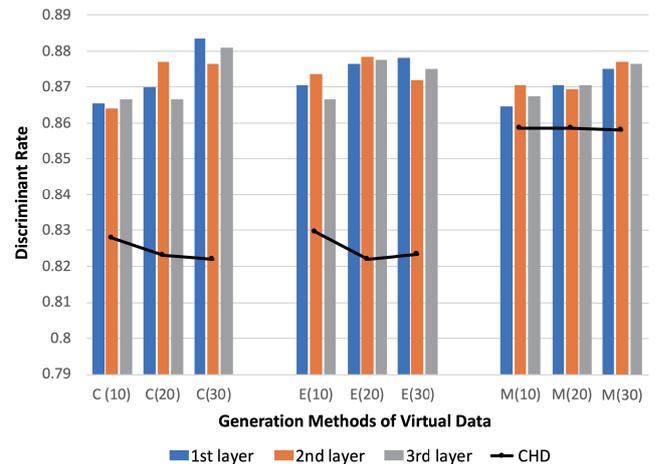


Fig. 4: Discriminant Rate at Each Layer

3種類のエポック回数における正識別型(C)、誤識別型(E)、混合型(M)の各層での学習データに対する識別率と評価データに対する識別率を図4に示す。学習データの識別率は10試行の平均値である。学習データの識別率は棒グラフで表示し、評価データの識別率は折れ線グラフで表示する。また、エポック回数(10, 30, 30)の混合型はM(30)として表している。図4から、正識別型、誤識別型、混合型の各層での識別率には次の特性があることがわかる。

- 1) 正識別型、誤識別型、混合型とも、学習データの識別率はエポック回数の増加にしたがって高くなる。評価データの識別率は混合型が最も高い。
- 2) 正識別型は、学習データに対する識別率は高い。しかし、層数が増加しても必ずしも安定的に識別率が高くなるとは限らない。評価データに対する識別率はエポック回数の増加に対して減少している。

Table 1: Comparison of Discrimination Rates among Three Methods

エポック回数	正識別型 (% , 個数)				誤識別型 (% , 個数)				混合型 (% , 個数)			
	識別率	正識別	誤識別	クラス替え	識別率	正識別	誤識別	クラス替え	識別率	正識別	誤識別	クラス替え
(10, 10, 10)	82.8	165.6	34.4	4.5	83.0	165.9	34.1	6.7	85.9	171.7	28.3	8.0
(10, 20, 20)	82.3	164.6	35.4	4.2	82.2	164.4	35.6	4.9	85.7	171.3	28.7	9.1
(10, 30, 30)	82.2	164.4	35.6	6.1	82.4	164.7	35.3	5.5	85.8	171.6	28.4	8.6
平均	82.4	164.9	35.1	4.9	82.5	165.0	35.0	5.7	85.8	171.5	28.5	8.6

学習データの過学習が生じている可能性が高い。

- 3) 誤識別型は、エポック回数や層数が増加しても学習データの識別率に概ね変化がない。評価データに対する識別率はエポック回数の増加に対して若干減少する。
- 4) 混合型は、エポック回数の増加に対して学習データの識別率が高くなり、層数の増加に対しても識別率は上昇している。評価データの識別率はエポック回数の増加に対しても安定している。過学習が抑えられている。

正識別型や誤識別型では、境界領域のバーチャルデータによって、学習データの識別精度は層数を重ねるごとに高くなる。しかし、学習データ数が膨大になれば、過度の学習によって、評価データの識別率は低下する。一方、混合型では、正識別型と誤識別型を交互に繰り返しながらバーチャルデータを発生するので、一方のクラスに過学習となっている識別線を他クラス方向に引き戻すことができ、学習データに対する識別率は安定し、評価データに対する識別率も高い。

各エポック回数に対する正識別型、誤識別型、混合型の評価データに対する識別率と学習データにおけるクラス推定でのクラス替えの結果を表1に示す。ただし、識別率と個数は10試行の平均値である。評価データに対する識別率の比較では、混合型は、正識別型より3.1%, 3.4%, 3.6%と高く、平均でも3.4%ほど高い。また、誤識別より2.9%, 3.5%, 3.4%と高く、平均でも3.3%ほど高い。一方、クラスを変更したバーチャルデータの個数は、混合型が最も多く、正識別型や誤識別型と比較してほぼ2倍となっている。混合型が、正識別型や誤識別型よりも識別線がより大きく変化し、バーチャルデータのクラスが頻繁に付け替えられていることがわかる。

以上、数値データが必ずしも十分な質と量とはいえないが、これらの解析結果から、バーチャルデータの発生方法に関して、次の点が明らかになった。

- ここでの数値例では、pdi-Baggingのバーチャルデータの発生方法で最も有効的な方法は混合型となった。混合型は、正識別型と誤識別型を交互に繰り返しながらバーチャルデータを発生するので、過学習となっている識別線を引き戻すことができ、学習データと評価データに対する識別率が高い。
- 混合型での多数決による評価データの識別率は良

好となった。pdi-Baggingのアルゴリズムが構造的にも良好に機能していることがわかる。

- バーチャルデータのクラス付け替えの結果において、混合型の識別線が大きく変動し、識別率が上昇していることから、クラス推定アルゴリズムが有効であることがわかる。

4. おわりに

本論文では、pdi-Baggingのバーチャルデータの発生方法について議論した。数値例から、バーチャルデータの発生法の特徴を明らかにし、最も効果的な発生方法が混合型である結果を得た。また、バーチャルデータのクラス付け替えのアルゴリズムが有効に機能していることを確認した。しかし、本論文での数値データが必ずしも十分な質と量とはいえないので、今後、様々な質と量の数値データを用いてバーチャルデータの発生方法の特徴を検証するとともに、実際のパターン分類の事例に応用して、本手法の有用性を検証する必要がある。

5. 謝辞

本研究を遂行するにあたり、日立造船(株)の藤吉誠氏、(株)ニチゾウテックの堅多達也氏にご助言を頂きました。ここに厚く感謝の意を表します。

参考文献

- [1] 上田：アンサンブル学習，情報処理学会論文誌，Vol.46, No.SIG15(CVIM12)，pp.11-20 (2005)
- [2] 村田，金森，竹ノ内：ブースティングと学習アルゴリズム：三人寄れば文殊の知恵は本当か？電子情報通信学会誌，Vol.88, No.9, PP.724-729 (2005)
- [3] 金森，畑埜，渡辺：ブースティング，森北出版 (2006)
- [4] 中島，荘司：ファジィアンサンブル識別器構築におけるデータ分割の効果，第25回ファジィシステムシンポジウム講演論文集，No.3E2-01 (2009)
- [5] 林，鶴背：確率的データ補間を用いたBCIのためのBoostingアルゴリズムの提案，信学技報，Vol.109, No.461, pp.303-308 (2010)

- [6] 林, 鶴背: 脳コンピュータインタフェースのための pdi-Boosting の提案, 第 26 回ファジィシステムシンポジウム講演論文集, pp.288-291 (2010)
- [7] I.Hayashi, S.Tsuruse, J.Suzuki, R.T.Kozma: A Proposal for Applying pdi-Boosting to Brain-Computer Interfaces, *Proc. of 2012 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE2012) in 2012 IEEE World Congress on Computational Intelligence (WCCI2012)*, pp.635-640 (2012).
- [8] I.Hayashi, S.Tsuruse: An Evaluation of pdi-Boosting for Brain-Computer Interfaces, *Proc. of the 6th International Conference on Soft Computing and Intelligent Systems and the 13th International Symposium on Advanced Intelligent Systems (SCIS-ISIS2012)*, pp.1215-1220 (2012).
- [9] 市橋, 渡邊: 簡略ファジィ推論を用いたファジィモデルによる学習型制御, 日本ファジィ学会誌, Vol.2, No.3, pp.429-437 (1990)

[連絡先]

林 勲 関西大学大学院 総合情報学研究科
〒 569-1095 大阪府高槻市霊仙寺町 2-1-1
tel. 072-690-2448
fax. 072-690-2491
e.mail ihaya@cpii.kutc.kansai-u.ac.jp