

# Bagging と Boosting を統合したアンサンブル学習手法

## An Ensemble Learning Method Integrating Bagging and Boosting

安村 禎明                      上原 邦昭  
Yoshiaki YASUMURA        Kuniaki UEHARA

神戸大学 工学部 情報知能工学科

Department of Computer and Systems Engineering, Faculty of Engineering, Kobe University

This report presents a new ensemble learning method IBB (Integration of Bagging and Boosting). This method creates initial base classifiers by bagging, then builds new base classifiers by boosting using the previous created base classifiers. This method has a new reweighting technique in boosting. This technique increases the weight of a sample when the sample is misclassified both the ensemble classifier and each base classifier. The experimental results using the datasets of UCI machine learning repository show that IBB results better accuracy than the other ensemble learning methods in several datasets.

### 1. はじめに

近年、機械学習の分野においてアンサンブル学習が注目を集めている。アンサンブル学習では、既存の学習アルゴリズムを用いて複数のベース分類器を生成し、そのベース分類器の結果を統合することで最終的な分類を行う。アンサンブル学習の著名な手法として Boosting [Freud 95] と Bagging [Breiman 96] がある。Boosting は一般に Bagging よりも高い精度が得られることが知られている。しかし、Boosting よりも Bagging の方がノイズに対して強いという特性がある。

このようにそれぞれの手法には利点と欠点があるため、これら 2 手法を統合することによって、さらに優れた手法の開発が試みられてきた。MultiBoost [Webb 00] は、Boosting と Bagging の変種である Wagging を組み合わせた手法である。この手法では、Wagging によって複数のベース分類器を作成し、そのベース分類器を元に Boosting を行う。Parallelized Boosting [Yu 01] は Bagging によって複数のベース分類器を作成し、誤分類したベース分類器の割合を用いて次のサンプル重みを調整し、Boosting を行う。これらの手法でのサンプル重みの調整は、MultiBoost では個々のベース分類器の分類結果のみを用いており、Parallelized Boosting では分類器全体の分類結果のみを用いている。

本研究では、個々のベース分類器と分類器全体の分類結果の両方を用いることで、サンプル重みの調整を行なう Bagging と Boosting を統合したアンサンブル学習手法を提案する。この手法を IBB アルゴリズム (Integration of Bagging and Boosting) と呼ぶ。本手法では、まず Bagging によって初期のベース分類器を作成し、そのベース分類器を基に Boosting を行なう。最終的な分類は、生成されたすべてのベース分類器の投票によって得られる。

### 2. アンサンブル学習

ここでは、様々なアンサンブル手法について概観する。Boosting は分類が困難なサンプルに対してその重みを大きくすることで分類精度の向上を図った手法である。Boosting では、まずすべてのサンプル重みを等しくしたデータを用いて既存の学習手法によってベース分類器を生成する。次に、前回に生成した分類器が誤分類したサンプルの重みを大きくし、そのデータ

連絡先: 安村 禎明, 神戸大学工学部情報知能工学科,  
E-Mail: yasumura@ai.cs.kobe-u.ac.jp

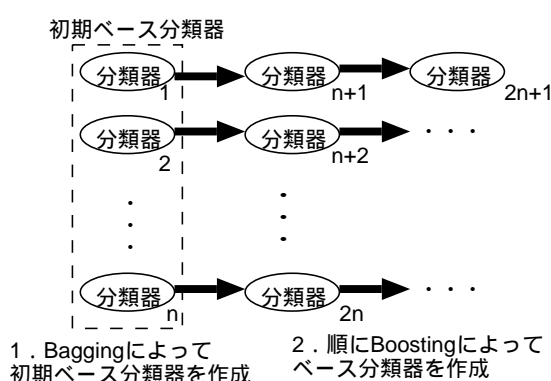


図 1: IBB アルゴリズムの概要

を用いて更にベース分類器を作成する。これを繰り返すことで複数のベース分類器を作成し、最終的な分類はそれらの投票で得られる。

Bagging は、データセットに対してサブサンプリングを行い、異なる複数のデータセットを作成し、それぞれに対して分類器を作成する。最終的な分類はその (重み付き) 投票によって得られる。Wagging [Bauer 99] は Bagging の変種である。Bagging ではランダムサンプリングを用いているが、Wagging はランダム重み付けを用いている。つまり、Wagging ではすべてのサンプルを用いるが、サンプルの重みをランダムに設定してベース分類器を作成する。一般に Boosting の方が Bagging よりも分類精度が良いことが多いが、Bagging の方が Boosting よりもノイズに対して強いことが知られている [Quinlan 96, Bauer 99]。

MultiBoosting は、Boosting と Wagging を組み合わせた手法である。この 2 手法を組み合わせることで、Boosting の精度向上と Wagging のノイズに対する強さを併せ持ったアンサンブル学習手法の作成を試みている。この手法では、Wagging によって初期ベース分類器を作成し、その初期ベース分類器をもとに Boosting することで更にベース分類器を生成する。したがって、初期のサンプル重みが異なるデータセットをもとにそれぞれで Boosting によってベース分類器の系列を作成し、それらを統合したものが最終的な分類結果となる。

Parallelized Boosting は、Boosting の並列化を狙った手法

**Algorithm IBB**

**Given:**  $S = \langle (x_1, y_1), \dots, (x_m, y_m) \rangle$   
 where  $x_i \in X, y_i \in Y = \{1, \dots, k\}$

**For**  $t = 1, \dots, n$ :

1. Set  $S_t$  by random sampling with random weight
2.  $h_t = \text{BaseLearn}(S_t)$ .
3. Calculate the error of  $h_t$ :

$$\epsilon_t = \frac{\sum_{i: h_t(x_i) \neq y_i} \text{weight}_t(x_i)}{m}$$

set  $\beta_t = \epsilon_t / (1 - \epsilon_t)$ .

**For**  $t = n + 1, \dots, T$ :

1.  $S_t = S_{t-1}$
2. For each  $x_i \in S_t$ ,  
 if  $h_t(x_i) \neq y_i$  AND  $h_t^{\text{fin}}(x_i) \neq y_i$   
 then  $\text{weight}_t(x_i) = \text{weight}_t(x_i)$   
 otherwise  $\text{weight}_t(x_i) = \text{weight}_t(x_i) \times \beta_t$   
 standardize  $S_t$  to sum to  $m$ .
3.  $h_t = \text{BaseLearn}(S_t)$ .
4. Calculate the error of  $h_t$ :

$$\epsilon_t = \frac{\sum_{i: h_t(x_i) \neq y_i} \text{weight}_t(x_i)}{m}$$

set  $\beta_t = \epsilon_t / (1 - \epsilon_t)$ .

**Output** the final hypothesis:

$$h^{\text{fin}}(x) = \arg \max_{y \in Y} \sum_{t: h_t(x)=y} \log \frac{1}{\beta_t}$$

図 2: IBB アルゴリズム

である。この手法ではまず、Bagging によって初期ベース分類器を生成する。この処理は並列に行われ、これを 1 ラウンドとする。次のラウンドでは前回のラウンドでのベース分類器のうち、サンプルを誤分類した分類器の数が閾値よりも多い場合にそのサンプルの重みを大きくする。これを繰り返すことによって複数のベース分類器を作成する。

### 3. IBB アルゴリズム

本節では、IBB アルゴリズムについて述べる。このアルゴリズムは、基本的には Bagging と Boosting を組み合わせたものとなっている。従来の Bagging と Boosting を組み合わせた手法との最も大きな違いは重み更新方法である。

$T$  個のベース分類器を作成する IBB アルゴリズムの流れは以下の通りである (図 1)。

1. 初期ベース分類器を Bagging によって  $n$  個生成する。
2. ベース分類器が生成された順にそのデータセットを用いて Boosting を行なう。例えば、 $t$  回目の繰り返しでは、 $t$  番目に作成されたベース分類器をもとに Boosting を行なう。これを  $T - n$  回繰り返す。
3. 生成されたすべてのベース分類器の投票で最終的な分類結果を得る。

アルゴリズムの各過程での詳細を示す。まず、初期ベース分類器の生成方法について述べる。アンサンブル学習アルゴリズムにおいてベース分類器の多様性が重要であることが知られている。このため、本手法では多様なベース分類器を生成するた

表 1: 実験データの概要

| 名前            | サンプル数 | クラス数 | 属性 |    |
|---------------|-------|------|----|----|
|               |       |      | 連続 | 離散 |
| balance-scale | 625   | 3    | 4  | -  |
| breast-cancer | 286   | 2    | -  | 9  |
| credit-a      | 690   | 2    | 7  | 8  |
| credit-g      | 1000  | 2    | 7  | 13 |
| diabetes      | 768   | 2    | 8  | -  |
| glass         | 214   | 6    | 9  | -  |
| heart-c       | 303   | 2    | 6  | 7  |
| lymph         | 148   | 4    | 3  | 15 |
| vehicle       | 846   | 4    | 18 | -  |

めに、ランダムサンプリングで得られたサンプル集合に対して更にランダム重み付けを行なう。このようにして得られたデータセットを元に初期ベース分類器を生成する。

次に、IBB アルゴリズムでのサンプル重みの更新方法を説明する。MultiBoost で用いられているサンプル重みの更新方法は、一般的な Boosting と同様に個々のベース分類器によって誤分類されたサンプルに対して重みを大きくする。一方、Parallelized Boosting では、多くのベース分類器によって誤分類されたサンプルに対して重みを大きくする。したがって、MultiBoost ではローカルな情報のみを利用して重みを更新しており、逆に Parallelized Boosting ではグローバルな情報のみを用いて重みを更新している。IBB アルゴリズムでは、ローカルな情報とグローバルな情報の両方を利用する。具体的には、個々のベース分類器によって誤分類され、かつその時点でのベース分類器全体によって誤分類されたサンプルの重みを大きくしている。これは、個々のベース分類器によって誤分類されたとしても分類器全体として正しく分類できれば問題ないため、このような重み更新方法をとった。以上をまとめると IBB アルゴリズムは図 2 のようになる。

### 4. 実験

#### 4.1 実験設定

本手法の有効性を検証するために他のアンサンブル学習手法との比較実験を行った。本実験では、表 1 に示す UCI repository の 9 種類のデータセットを用いた。この表では、データセットの名前、サンプル数、クラス数、連続値と離散値をとる属性の数を示している。実験では、それぞれのデータセットに対して 10-fold cross-validation を 10 回行った。すべてのアンサンブル学習手法においてベース学習器は C4.5 を用いた。IBB アルゴリズムにおいて Bagging で生成するベース学習器は 10 個とした。これは、Bagging においてベース学習器を 10 個以上としても精度があまり変化しないことが実験的にわかったからである。

#### 4.2 実験結果と考察

表 2 は 9 つのデータセットに対する各学習手法の精度をまとめたものである。C4.5 以外はすべてベース分類器の数を 30 個とした。IBB は 9 つのデータセット中、balance-scale, credit-g, diabetes, heart-c の 4 つのデータセットで最も良い精度となった。これは比較した学習手法の中では最も多くなっており、IBB の精度の良さを表している。しかしながら、すべて

表 2: アンサンブル学習手法の精度比較

| Data Set      | C4.5  | Bagging | AdaBoostM1 | MultiBoost | Parallelized Boost | IBW   |
|---------------|-------|---------|------------|------------|--------------------|-------|
| balance-scale | 77.87 | 77.21   | 82.24      | 82.12      | 78.39              | 83.07 |
| breast-cancer | 73.53 | 73.49   | 65.73      | 72.34      | 73.12              | 70.83 |
| credit-a      | 85.32 | 81.94   | 82.89      | 83.21      | 84.48              | 83.57 |
| credit-g      | 71.34 | 74.18   | 74.40      | 75.66      | 73.20              | 75.79 |
| diabetes      | 74.22 | 76.23   | 62.63      | 76.27      | 75.59              | 76.34 |
| glass         | 67.01 | 73.97   | 77.10      | 76.96      | 72.32              | 74.20 |
| heart-c       | 76.11 | 81.41   | 79.53      | 81.58      | 80.99              | 83.10 |
| lymph         | 76.76 | 80.47   | 84.45      | 83.98      | 79.42              | 83.10 |
| vehicle       | 72.21 | 72.73   | 77.06      | 75.35      | 74.18              | 74.47 |
| average       | 74.93 | 76.85   | 76.23      | 78.61      | 76.85              | 78.29 |

のデータセットでの精度の平均では、MultiBoost が最も良くなった。IBW は一部のデータセットであまり精度の良くないものがあつたことが原因と考えられる。特に breast-cancer と glass データセットにおいて MultiBoosting よりも精度が悪くなった。

次に、各データセットに対する実験結果の詳細を考察する。図 3 は heart-c データセットに対して Boosting, Bagging, MultiBoost, IBB を適用した場合のベース学習器の数と精度との関係を示したグラフである。MultiBoost と IBB はベース分類器を 10 個作成したところから始まるため、このグラフでも 10 のところからあらわれている。ベース分類器の数が増加するにつれて精度が向上することがすべての学習手法で見られるが、ある数以上になるとあまり精度が上がらなくなっている。このデータセットでは、IBB はほぼすべての場合において最も良い精度を記録した。

図 4 は breast-cancer データセットに対して各学習手法を適用した場合の精度変化を示している。このデータセットでは、Bagging 以外の手法はベース分類器の数が増加するにつれて分類精度が下がっている。すなわち、誤分類に基づくサンプル重みの更新を行っているアルゴリズムがその繰り返しによって精度が低下している。IBB や MultiBoosting では、最初に Bagging や Wagging を行うため、ベース分類器が少ないうちは精度低下はあまりないが、数が増加するにしたがって精度が下がっている。MultiBoost よりも IBB の方が精度低下が大きく、これが表 2 での差となつたと考えられる。

図 5 は glass データセットに対して各学習手法を適用した場合の精度変化を表している。このデータセットでは、IBB は Boosting や MultiBoost よりも精度が低くなつた。このようにデータセットによっては IBB の精度が他の学習手法よりも良くない場合もある。このようなときにデータセットに対応して手法を変更する枠組が必要となる。例えば、ベース分類器の数を制御することやサンプル重みの更新方法を変更するなどが考えられる。

#### 4.3 重み更新方法の考察

本手法で採用したサンプル重みの更新は Boosting に対しても適用することが可能である。そこで、IBB でのサンプル重みの更新方法を用いた Boosting(便宜上、Boosting2 と呼ぶ)と一般的な Boosting を比較する。図 6 は glass と lymph データセットでこれら 2 手法を比較したグラフであり、ベース分類器の数とその精度の関係を示している。Boosting はベース分類器の数が増加すると多くの場合精度も向上しており、大きく

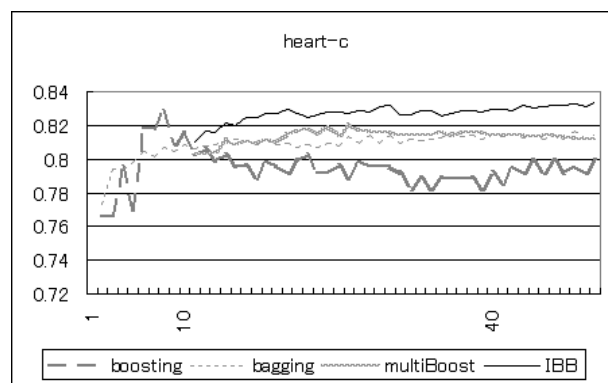


図 3: heart-c データセットに対する学習手法の精度比較

精度が下がることはなかった。しかし、Boosting2 では、ある時点から大きく精度が下がった。これは他のデータセットに関しても同様の傾向が見られた。

この原因について考察する。IBB の重み更新方法を Boosting に適用すると、ベース分類器の数が増加するにしたがって、分類器全体としては同じサンプルを誤分類するようになるため、繰り返しを行っても同様の重みを持ったサンプルがでやすくなる。このため、ベース分類器の多様性が失われ、精度の低下をまねいたと考えられる。一方、IBB ではこの重み更新方法がうまく機能していることが多い。IBB では Bagging を用いて多様性を実現しているため、このような低下が起こりにくいと考えられる。しかし、ベース分類器の数が増加すると同様の現象が起こる可能性がある。このため、ベース分類器の数を制御することを検討する必要がある。

## 5. おわりに

本稿では、Bagging と Boosting を統合した IBB アルゴリズムを提案した。IBB では、サンプル重みの更新のための新たな方法を用いている。これは個々のベース分類器と分類器全体が誤分類したサンプルに対して重みを大きくするものである。本手法と他のアンサンブル学習手法と比較した結果、いくつかのデータセットにおいて最も良い精度が得られた。しかしながら、データセットによっては本手法の精度が低い場合があり、そのようなデータセットに対して繰り返し回数や重みの更

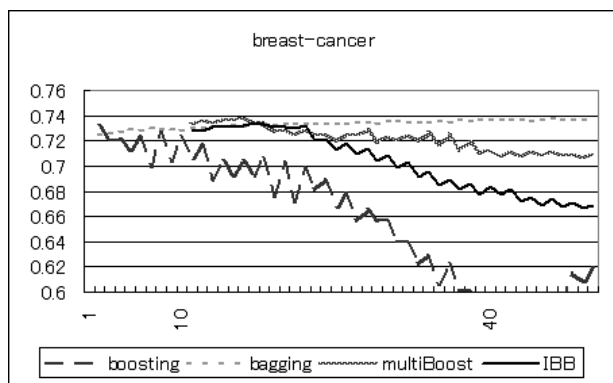


図 4: breast-cancer データセットに対する学習手法の精度比較

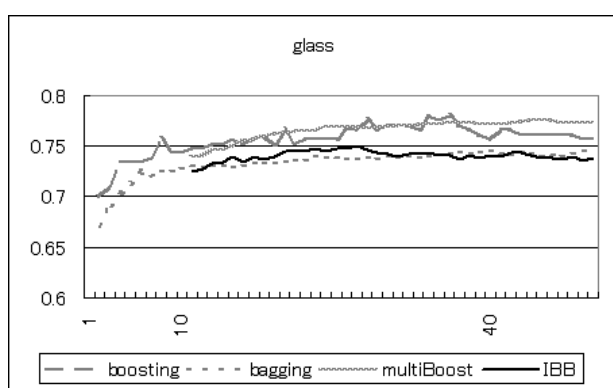


図 5: glass データセットに対する学習手法の精度比較

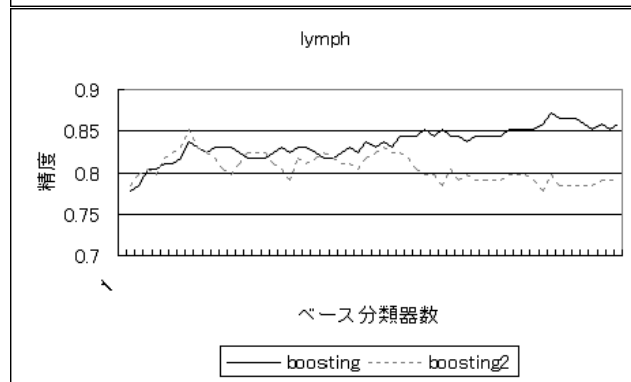
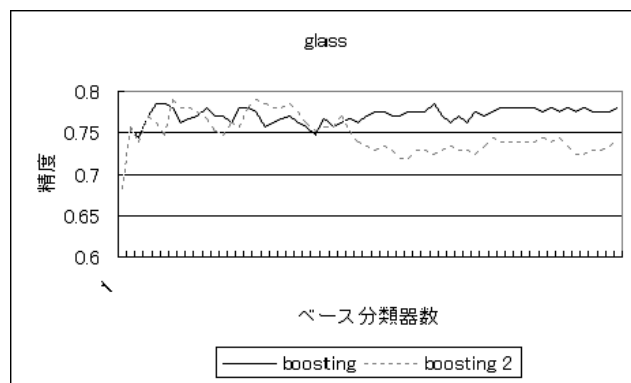


図 6: 重み更新方法の違いによる精度比較

Boosting, and Variants, Machine Learning, Vol.36, pp.105-142 (1999).

新方法を検討する必要がある。

### 参考文献

- [Freud 95] Freud, Y. and Schapire, R. E. : A decision-theoretic generalization of on-line learning and an application to boosting, Journal of Computer and System Sciences, Vol.55, pp.119-139 (1995).
- [Breiman 96] Breiman, L. : Bagging predictors, Machine Learning, Vol. 24, 123-140 (1996).
- [Webb 00] Webb, G. I. : MultiBoosting: A Technique for Combining Boosting and Wagging, Machine Learning, Vol. 40, pp.159-196 (2000).
- [Yu 01] Yu, C. and Skillicorn, D.B. : Parallelizing Boosting and Bagging, Technical Report 2001-442, Department of Computing and Information Science, Queen's University (2001).
- [Quinlan 96] Quinlan, J. R. : Bagging, boosting, and C4.5, Proceedings of the Thirteenth National Conference on Artificial Intelligence, pp.725-730, (1996).
- [Bauer 99] Bauer, E. and Kohavi, R : An Empirical Comparison of Voting Classification Algorithms: Bagging,