

## 相補的な学習機構に基づく協調型アンサンブル学習法

Collaborative Ensemble Learning based on Complementary Learning Framework

野宮 浩揮\*1 上原 邦昭\*2  
Hiroki Nomiya Kuniaki Uehara\*1神戸大学大学院自然科学研究科  
Graduate School of Science and Technology, Kobe University\*2神戸大学大学院工学研究科  
Graduate School of Engineering, Kobe University

Constructing and integrating multiple classifiers based on diverse classification algorithms leads to an efficient classification which can be flexibly applied to various classification problems. This strategy is known as ensemble learning. However, it is difficult to determine the appropriate classifier because of the lack of interaction between multiple classifiers. In addition, the performance of a classifier is often deteriorated due to overfitting caused by hard examples which are difficult to correctly classify. In order to solve these problems, we propose an effective ensemble learning framework by introducing the interaction between multiple classifiers. Through the learning process, the classifiers interact with each other so that they can complement each other. As a result, we can determine the appropriate classifier according to given data and prevent overfitting by eliminating hard examples from the training sets.

## 1. はじめに

複数の分類器を統合し、より性能の高いアンサンブル分類器を構成するアンサンブル学習法が広く知られている [Dietterich 00]. しかし、既存のアンサンブル学習法には、以下に述べる問題点のため、分類精度の低下が起こることがある。一つ目の問題点として、データによって分類に適した学習アルゴリズムが異なるという点がある。例えば、図 1 に示すデータを分類する問題を考える。図中の○は正事例、×は負事例を表している。また、図 1 (a) と (c) は決定木アルゴリズムを用いた分類器、(b) と (d) は線形判別アルゴリズムを用いた分類器による学習結果 (生成された識別境界) を表している。

図 1 (a) と (b) は同じ分布を持つデータであるが、決定木アルゴリズムによる識別境界を用いると、すべての事例は正しく分類されるのに対して、線形判別アルゴリズムによる識別境界では、多くの事例が誤分類されている。逆に、図 1 (c) と (d) のデータについては、決定木アルゴリズムによる識別境界では多数の事例が誤分類されているのに対して、線形判別アルゴリズムによる識別境界ではすべての事例が正しく分類されている。このように、各学習アルゴリズムには、正確に分類できるデータ分布とそうでないデータ分布がある。したがって、与えられたデータの分布に応じて、適切な学習アルゴリズム (分類器) を選択する必要がある。

もう一つの問題点として、分類が困難な事例を学習することによって、アンサンブル分類器の性能が逆に低下することがある。図 1 の例からもわかるように、分類に適さないデータを用いると性能の低い分類器が生成される。個々の分類性能が低いと、それらを統合しても精度の改善は見込めない。したがって、分類に適していないデータを排除し、適切なデータのみを用いてアンサンブル学習を行うことが望ましい。しかし、従来のアンサンブル学習法ではデータの性質を十分考慮していないため、正しく分類できるか否かにかかわらず、分類に適していないデータも含めて学習してしまう可能性がある。結果として

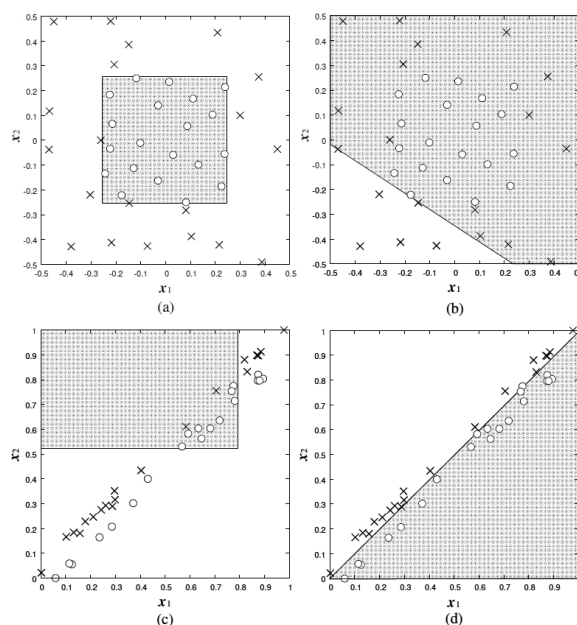


図 1: 決定木および線形判別アルゴリズムによる識別境界

て、アンサンブル学習器の分類性能は与えられたデータの分布に依存してしまうことになる。

本稿では、これらの問題を解決するために、協調的なアンサンブル学習法を提案する。複数の分類器が協調的に学習すれば、適切な学習アルゴリズムを選択的に用いてアンサンブル分類器を生成できる。特に、代表的な協調的学習機構として、次の三種類の学習機構を提案する。一つ目は、各々の分類器が互いに助け合う学習法であり、相補的学習機構と呼ぶ。二つ目は、分類器同士が互いに競い合う学習法であり、競合的学習機構と呼ぶ。三つ目は、分類器の間で合議を行って学習戦略を決定する学習法であり、合議的学習法と呼ぶ。以下の章では、これらの学習機構を用いた協調型アンサンブル学習法について述

べ、学習機構の差異が分類性能に対して及ぼす影響について、分類性能の比較実験を通じて検証する。

## 2. 協調型アンサンブル学習モデル

協調型アンサンブル学習では、正しく識別するのが困難な事例（これを hard example と呼ぶ）の検出が重要となる。例えば、図 1 (b), (c) は hard example が多い状態である。決定木アルゴリズムおよび線形判別アルゴリズムに基づく二種類の分類器  $X_D, X_L$  を用いて、hard example の例を図 2 に示す。

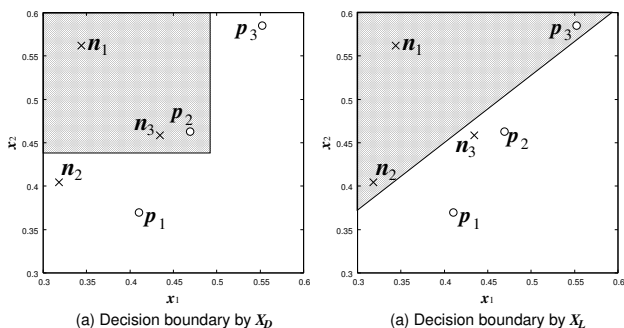


図 2: Hard example の例

図 2 (a), (b) は各々  $X_D, X_L$  の識別境界を表している。図中の正事例  $p_1$  と負事例  $n_1$  は、 $X_D$  と  $X_L$  の双方により誤分類される。このような事例を本質的な hard example と呼ぶ。本質的な hard example とは、全ての分類器により誤分類される事例を指す。一方、正事例  $p_2$  と負事例  $n_2$  は、 $X_L$  によってのみ誤分類される。逆に、正事例  $p_3$  と負事例  $n_3$  は、 $X_D$  によってのみ誤分類される。このように、一部の分類器によってのみ誤分類される事例を分類器依存の hard example と呼ぶ。

本質的な hard example は、いずれの分類器によっても正しく分類されないため、ノイズ事例や外れ値を持つ事例など、学習対象から除外すべき事例である可能性が高い。一方、分類器依存の hard example は、適切に用いれば学習に役立てることができる。しかし、従来のアンサンブル学習法では、分類器を別々に訓練するので、本質的な hard example と分類器依存の hard example を区別できず、学習効率の低下につながっている。そこで、本質的な hard example を除去し、分類器依存の hard example を適切に利用するために、図 3 に示すような協調型アンサンブル学習モデルを提案する。

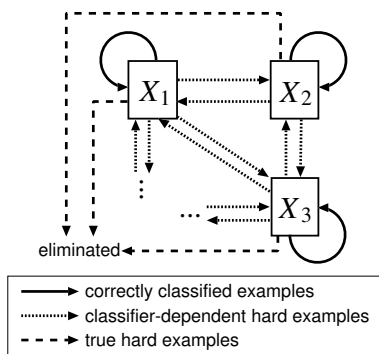


図 3: 協調型アンサンブル学習モデル

本モデルでは、任意の数の分類器を同時に訓練できる。例え

ば、図 3 において、分類器  $X_1, X_2$  は前述の分類器  $X_D, X_L$  に相当している。さらに、必要に応じて  $X_3, X_4, \dots, X_N$  と分類器を追加することができる。本質的な hard example と分類器依存の hard example を区別するために、例えば  $X_1$  は他の分類器 ( $X_2, X_3, \dots, X_N$ ) に対して、自身がどの事例を誤分類したかを通知する。逆に  $X_1$  以外の分類器は、誤分類した事例を  $X_1$  に通知する。そして、各分類器は通知された事例を重点的に学習する。このように、各分類器が他の分類器に誤分類した事例を通知すると、分類器依存の hard example を適切に用いて協調的に学習できる。また、いずれの分類器でも誤分類された事例は本質的な hard example であるとみなされ、すべての分類器の訓練集合から除去される。こうして協調的な訓練を行うと、hard example による過学習を抑制し、より効率的なアンサンブル学習を行えるようになる。

## 3. 協調的アンサンブル学習アルゴリズム

### 3.1 Hard example 検出のための重み付け

本章では、協調的アンサンブル学習の定式化を行う。まず、 $m$  を訓練事例の数とすると、訓練集合  $S$  は  $S = \{(x_1, y_1), \dots, (x_m, y_m)\}$  と表される。 $x_i$  は単一の事例、 $y_i$  はクラスラベルを表しており、 $C$  クラスの分類問題では  $y_i \in \{1, \dots, C\}$  となる。また、各事例は分類の難しさを表す重みを持つ。提案手法では、重みを用いて反復的に学習を行う（この学習の反復をラウンドと呼ぶ）。具体的には、各ラウンドで事例の重みに応じて訓練事例のサンプリングを行い、重みの大きい事例を重点的に学習する。これは、AdaBoost アルゴリズム [Freund 97] と同様である。しかし、AdaBoost は hard example を検出できず、hard example によって過学習を起こしやすい。そこで、本質的な hard example と分類器依存の hard example を検出できる重み付けアルゴリズムを提案する。まず、訓練事例のサンプリングに用いる事例の重みの分布を、次の二つの観点に基づいて定義する。

- 過学習を防ぐため、事例  $x$  が分類器  $X$  にとって hard example であると見なされた場合、 $X$  の訓練データのサンプリング対象から  $x$  を除外する。
- 各分類器を相補的に訓練するため、分類器  $X$  が事例  $x$  を誤分類した場合、他の分類器により  $x$  がサンプリングされるようにする。

#### 3.1.1 相補的学習のための重み付け

前述の観点に基づき、相補的学習、競合的学習、合議的学習それぞれの学習機構に対する重み付け規則を構成する。まず、相補的学習のための重み付けについて述べる。相補的学習では、それぞれの分類器は、自身が正しく分類できない事例を他の分類器に学習してもらう代わりに、他の分類器が正しく分類できない事例を学習する。この学習法を定式化するために、第  $t$  ラウンドでの  $l$  番目の分類器  $X_t^l$  に対する重みの分布  $D_t^l(i)$  ( $i = 1, \dots, m$ ) を以下のように定義する。

$$D_t^l(i) = \frac{\delta_{i,t}^l d_{i,t}^l}{\sum_{i=1}^m \delta_{i,t}^l d_{i,t}^l} \quad (1)$$

ここで、 $\delta_{i,t}^l$  は  $x_i$  が  $X_t^l$  にとっての hard example であると見なされた場合 0、そうでなければ 1 となる。また、

$$d_{i,t}^l = \frac{1}{N} \sum_{j=1}^N \left( \frac{w_{i,t}^j}{\sum_{i=1}^m w_{i,t}^j} \right) \quad (2)$$

である。\$N\$ は分類器の数を表し、\$w\_{i,t}^l\$ は第 \$t\$ ラウンドでの \$X^l\$ に対する \$x\_i\$ の重みを表す。\$x\_i\$ が \$X^l\$ によって hard example であると見なされた場合、\$\delta\_{i,t}^l = 0\$ となり、式 (1) より \$D\_t^l(i) = 0\$ となる。このとき、\$x\_i\$ はサンプリングされず、この事例は \$X^l\$ の訓練集合から除外されることになる。ある事例が hard example かどうかは、その事例の重みが閾値を越えているかどうかによって判断される。具体的には、beam search 法を用いて最適（もしくは準最適）な重みの閾値を決定している [Nakamura 04].

式 (2) が相補的学習機構を表している。この式より、\$d\_{i,t}^l\$ は \$X^l\$ 以外の分類器に対する事例の重みに依存して決定されることがわかる。したがって、\$X^l\$ 以外の分類器によって誤分類された事例はサンプリングされやすくなり、結果的に \$X^l\$ は他の分類器が誤分類した事例を重点的に学習することになる。

### 3.1.2 競合的学習法のための重み付け

競合的学習法では、各分類器は他の分類器よりも可能な限り高い分類精度を実現することを目標として訓練される。具体的には、各ラウンドで重みの分布を求める際に、他の分類器が正しく分類できなかった事例のみに重みを与えることにより、他のすべての分類器が正しく分類した事例はサンプリングされないようにする。また、この事例を誤分類した分類器の数が多く、大きい重みが与えられる。そこで、式 (2) の代わりに以下の式を用いて重みの分布を求めている。

$$d_{i,t}^l = \frac{1}{N-1} \sum_{j \neq i}^N \left( \frac{k_{i,t}^j w_{i,t}^j}{\sum_{i=1}^m k_{i,t}^j w_{i,t}^j} \right) \quad (3)$$

ここで、\$k\_{i,t}^j\$ は \$x\_i\$ が \$X\_t^j\$ により正しく分類されれば 0、誤分類されれば 1 となるので、重みは \$x\_i\$ を誤分類した分類器の数に比例する。

### 3.1.3 合議的学習法のための重み付け

合議的学習法では、ラウンドが終了するごとにすべての分類器で合議を行い、次のラウンドで各分類器にどの訓練事例を割り当てるかを決定する形式で学習を行う。具体的には、まず各訓練事例について、各分類器に対するクラス分離度を求める。クラス分離度とは、あるクラスに属する事例をどれだけ正確に分類できるかを表す指標であり、アンサンブル分類器の予測を決定する際に用いる。詳細な定式化は後述する。そして、クラス分離度の高い上位 \$\nu\$ 個の分類器に対してのみ、次のラウンドでその事例を訓練事例として与える。このため、式 (1) の代わりに以下の式を用いて重みの分布を決定している。

$$D_t^l(i) = \frac{\kappa_{i,t}^l \delta_{i,t}^l d_{i,t}^l}{\sum_{i=1}^m \kappa_{i,t}^l \delta_{i,t}^l d_{i,t}^l}. \quad (4)$$

ここで、\$\kappa\_{i,t}^l\$ は \$x\_i\$ に対するクラス分離度の値が \$N\$ 個の分類器の中で上位 \$\nu\$ 番目以内であれば 1、そうでなければ 0 となる。

## 3.2 アンサンブル学習器の生成

学習のラウンド数を \$T\$ とすると、各分類アルゴリズムに対して、\$T\$ 個の分類器 \$\{X\_t^l\}\_{t=1}^T\$ を統合してアンサンブル分類器 \$X^l\$ を構成する。その際、重み付き投票法を用いて、すべての分類器 \$\{X\_t^l\}\$ の予測を統合した結果を \$X^l\$ の予測と定める。事例 \$x\$ に対する \$X^l\$ の予測 \$X^l(x)\$ は、各分類器の分類性能 \$\alpha\_t^l\$ を重みとして、以下のように定義している。

$$X^l(x) = \operatorname{argmax}_{c \in \{1, \dots, C\}} \sum_{t=1}^T \alpha_t^l [X_t^l(x) = c]. \quad (5)$$

ここで、\$[X\_t^l(x) = c]\$ は、\$X\_t^l(x) = c\$ のとき 1、\$X\_t^l(x) \neq c\$ のとき 0 となる。また、\$\epsilon\_t^l\$ を \$X^l\$ の誤分類率とすると、\$\alpha\_t^l = \log \frac{1-\epsilon\_t^l}{\epsilon\_t^l}\$ である。すなわち、\$\alpha\_t^l\$ の値は誤分類率に反比例する。

さらに、ある事例が hard example かどうかを判断し、hard example でない場合のみ、その予測を信頼するための指標として、クラス分離度を定義する。クラス分離度は各クラスについて定義され、正確に分類できている事例の割合に比例する。Hard example は誤分類されやすいので、多く含まれるとクラス分離度は低くなる。ここで、\$X\_t^l\$ がある事例 \$x\$ のクラスラベルを \$c\$ と予測した場合を考える。このとき、もしクラス \$c\$ に対する \$X\_t^l\$ のクラス分離度が高ければ、\$x\$ は hard example でない可能性が高く、この予測は信頼できる。クラス \$c\$ に対する \$X\_t^l\$ のクラス分離度 \$s\_t^l(c)\$ は、以下のように定義される。

$$s_t^l(c) = \begin{cases} s_{t+}^l(c) & \text{if } X_t^l(x) = c \\ s_{t-}^l(c) & \text{otherwise.} \end{cases} \quad (6)$$

ここで、

$$X_t^l(x) = \operatorname{argmax}_{c \in \{1, \dots, C\}} \sum_{\tau=1}^t \alpha_\tau^l [X_\tau^l(x) = c]$$

$$s_{t+}^l(c) = \frac{n_{t,c,c}^l}{\sum_{i=1}^C n_{t,c,i}^l}, \quad s_{t-}^l(c) = \frac{\sum_{i \neq c}^C \sum_{j \neq c}^C n_{t,i,j}^l}{\sum_{i \neq c}^C \sum_{j=1}^C n_{t,i,j}^l}$$

である。また、\$n\_{t,i,j}^l\$ は、クラス \$i\$ に属しており、クラス \$j\$ に分類された事例数を表す（\$i = j\$ ならば正しく分類されており、\$i \neq j\$ ならば誤分類されている）。\$s\_{t+}^l(c)\$ は、クラス \$c\$ に属する事例の多くが正しく分類されると高くなり、\$s\_{t-}^l(c)\$ の値は、クラス \$c\$ に属さない事例がクラス \$c\$ に誤分類されなければ高くなる。例えば、分類器 \$X\_t^l\$ がクラス \$c\$ に属する事例の大部分を正しく分類できれば、クラス \$c\$ に対するクラス分離度が高くなる。\$X\_t^l\$ がクラス \$c\$ に属していない事例の多くをクラス \$c\$ に誤分類した場合は、クラス \$c\$ に対するクラス分離度が低くなる。アンサンブル分類器 \$X\_t^l(x)\$ の予測したクラスラベルが \$c\$ であれば、\$s\_{t+}^l(c)\$、\$c\$ でなければ、\$s\_{t-}^l(c)\$ がクラス分離度として用いられる。

最終的に、クラス分離度に基づいて、アンサンブル分類器の信頼度が評価される。未知の事例を分類する際に、まずすべてのアンサンブル分類器 \$\{X\_t^l\}\_{t=1}^N\$ の予測結果を得る。そして、各アンサンブル分類器のクラス分離度を求め、クラス分離度が最大となるアンサンブル分類器をその予測を最終予測として出力する。したがって、事例 \$x\$ に対する最終予測 \$F(x)\$ は、以下のように定められる。

$$F(x) = X^{l^*}(x) \text{ such that } l^* = \operatorname{argmax}_l \sum_{\tau=1}^t s_\tau^l(X_\tau^l(x)). \quad (7)$$

## 4. 実験

### 4.1 提案手法の分類性能の検証

提案手法の性能を評価するため、UCI Machine Learning Repository [Asuncion 07] より取得したデータセットを用いて、分類精度の比較実験を行った。提案手法では、SVM, C4.5, Naive Bayes (NB), Logistic Regression (LR) アルゴリズムを統合してアンサンブル分類器を構築した。比較対象として、上記 4 種類の学習アルゴリズムをそれぞれ単体で用いて AdaBoost

を実行した。また、ラウンド数はいずれも実験的に 10 に設定した。5-fold の交差検定を用いた実験結果を表 1 に示す。表中の下線は有意水準 5% における  $t$  検定を行った結果、提案手法との間に有意な差が見られたことを示している。

表 1: 各分類器の分類精度 (%)

	AdaBoost				提案手法
	SVM	C4.5	NB	LR	
glass	89.3	97.2	94.4	94.9	97.7
ionosphere	87.2	92.6	92.6	85.8	93.2
iris	95.3	93.3	96.0	96.7	95.3
optdigits	98.1	97.5	91.3	96.1	98.3
pendigits	98.2	98.7	85.8	95.5	98.9
segment	92.5	97.8	80.3	94.9	97.3
spambase	91.2	94.8	79.5	92.6	95.1
wdbc	97.0	96.3	96.0	95.6	97.4
wine	96.6	98.3	97.2	97.2	98.3
yeast	55.6	56.3	56.9	59.1	58.7

表 1 より、多くのデータセットについて、提案手法と既存の学習アルゴリズムとの間に有意な差が見られる。この結果から、各分類器が相補的に学習を行うことにより、分類器を別々に訓練する場合に比べて、より効率的な学習ができていたことが確認できる。また、提案手法は与えられた事例に応じて適切な分類器を選択できるので、ionosphere や segment のように、用いる学習アルゴリズムによって分類精度が大きく異なる場合でも、高い分類精度を達成できている。

#### 4.2 Hard example に対する頑健性の検証

次に、hard example に対する提案手法の頑健性を検証するため、人工的に 20% のラベルノイズを混入したデータセットを用いて分類実験を行った。実験結果を表 2 に示す。

表 2: ノイズを含むデータに対する各分類器の分類精度 (%)

	AdaBoost				提案手法
	SVM	C4.5	NB	LR	
glass	63.6	72.9	53.7	60.8	76.6
ionosphere	69.8	70.4	72.7	68.1	78.6
iris	74.0	66.7	72.7	69.3	74.0
optdigits	76.5	73.0	72.2	74.7	77.4
pendigits	76.7	74.7	68.0	73.3	78.4
segment	73.6	71.2	55.6	72.6	76.3
spambase	72.3	69.0	68.4	73.8	73.4
wdbc	77.2	68.4	76.3	75.8	77.5
wine	78.7	71.4	73.6	73.0	75.8
yeast	43.8	42.4	44.7	46.3	47.2

ノイズの影響により、全体的に分類精度は著しく低下している。特に、Naive Bayes は glass, segment で、SVM や Logistic Regression は glass, ionosphere で、非常に大きく分類精度が低下している。この原因として、ノイズ事例に対する過学習が考えられる。一方、提案手法は、すべてのデータセットについて、著しい分類精度の低下は見られない。さらに、ノイズを含まないデータセットの実験結果と比較して、有意な差が見られるデータセットが増加している。このことから、提案手法が効

果的にノイズ事例を訓練事例から除外し、過学習を抑えていることが確認できる。

#### 4.3 分類器の相互作用による認識性能への影響

複数の分類器によるアンサンブル学習では、どのように分類器を相互作用させるかということが、協調型アンサンブル学習アルゴリズムを特徴づける要因となっている。そこで、前述の三種類の学習法を用いて比較実験を行った。合議的学習法では  $\nu$  の値を実験的に 3 としている。ノイズを含まない 5 つのデータセットに対して、各学習法を用いて実験を行った結果の分類精度を表 3 に示す。

表 3: 各学習法による分類精度 (%)

	glass	iris	spambase	wine	yeast
相補的	97.7	95.3	95.1	98.3	58.7
競合的	95.8	95.3	94.7	97.2	58.0
合議的	96.3	95.3	94.6	98.3	59.2

全体的には、相補的学習法が高い分類精度を達成しており、競合的学習法は他の学習法よりやや分類精度が低い。いずれのデータセットにおいても各学習法の間には有意な差は見られなかったが、多種の学習アルゴリズムを用いてアンサンブル学習を行う際には、各学習器が互いに協力的に学習することにより、学習アルゴリズムの多様性を生かす学習法が有効であると考えられる。また、有意な差はないものの、yeast では合議的学習法が相補的学習法を上回っている点から、いずれの分類器にとっても分類が難しい場合には、合議的な学習が有効である可能性がある。したがって、与えられたデータセットに応じて、適切な分類器を選択するだけでなく、適切な学習機構を選択することにより、さらなる分類性能の向上が期待できる。

## 5. まとめ

本稿では、多様な学習アルゴリズムを用いて複数の分類器を生成し、相補的に学習する協調型アンサンブル学習手法を提案した。提案手法は、Hard example を検出できる重み付け規則の導入により、適切な訓練データを用いて各学習器を訓練できる。実験を通じて、提案手法が高い分類性能を達成でき、hard example に対して頑健であることを示せた。今後の課題として、分類器間の相互作用のための学習手法を改善し、さらなる分類精度の向上を図ることが挙げられる。また、より効率的な学習機構を導入し、計算量を低減する必要がある。

## 参考文献

- [Asuncion 07] Asuncion, A. and Newman, D.: UCI Machine Learning Repository (2007)
- [Dietterich 00] Dietterich, T. G.: Ensemble Methods in Machine Learning, in *Proc. of 1st International Workshop on Multiple Classifier Systems*, pp. 1–15 (2000)
- [Freund 97] Freund, 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, No. 1, pp. 119–139 (1997)
- [Nakamura 04] Nakamura, M., Nomiya, H., and Uehara, K.: Improvement of Boosting Algorithm by Modifying the Weighting Rule, *Annals of Mathematics and Artificial Intelligence*, Vol. 41, pp. 95–109 (2004)