特　集　「計算学習理論の進展と応用可能性」

ブースティング入門

A Short Introduction to Boosting

ヨアブ・フロインド*1　ロバート・シャピリ*1　(訳：安倍　直樹)*2
Yoav Freund　Robert Schapire　Naoki Abe

*1 AT&T ラボラトリーズ リサーチ シャノン ラボラトリー
AT&T Labs – Research Shannon Laboratory, Florham Park NJ07932-0971, U.S.A.
*2 NEC C&C メディア研究所
NEC C&C Media Research Laboratories, Kawasaki 216-8555, Japan.

1999 年 7 月 13 日　受理

Keywords: boosting, adaboost, PAC-learning, computational learning theory, support vector machine.

1. はじめに

以下のような状況を考えよう．ある競馬ファンが，なるべく多くの配当を得ようと，各馬の過去の成績やオッズ等の情報に基づいて勝ち馬を予測するプログラムを作ろうとした．このようなプログラムを作るために，彼はまず熟練ギャンブラー (以下，エキスパート) にどのような戦略を用いているのかを説明してくれるように頼むことにした．ところが，彼は競馬は勘であって，説明できるような戦略などないと言う．しかし，具体的にいくつかのレース情報のリストを与えられると，このエキスパートは「最近の勝率の最も高い馬に賭けろ」とか「オッズの最も高い馬に賭けろ」などの経験則を問題なく見つけることができたという．確かにこのような経験則はおおざっぱであまり高い精度のルールとは言えないが，ただランダムに賭けているよりは少しはましな予測ができると思われる．また，エキスパートの意見をいくつもの異なるレース情報リストについて聞くことにより，競馬ファンは数多くの経験則を習得できる．

さて，こうして得られた経験則を上手に利用するには，競馬ファンは以下の二つの問題を解決しなくてはならない．一つめは，エキスパートに提示すべきレース情報リストの集合をどのように定めるかという問題であり，二つめは獲得された数多くの経験則をどのようにまとめて一つの精度の高いルールを得るかという問題である．「ブースティング」とは，このような設定の下，数多くの精度の低いルールを組み合わせて非常に精度の高い予測ルールを得るための，汎用的かつ理論的な性能保証のある方式である．この解説文では，ブースティングに関する最近の研究成果の中から，特にこれまで多くの理論的な検証と実験的実証がなされてきた AdaBoost というアルゴリズムを取り上げる．まず **3 章**で AdaBoost アルゴリズムを紹介し，**4〜7 章**でブースティングの理論的な基盤について説明する．ここでは，特にブースティングがなぜ「過学習」を避けられるかについても議論する．そして，**8 章**ではブースティングを用いた実験と応用について述べる．

2. 背　景

ブースティングのルーツは，機械学習方式を解析する理論的な枠組である「PAC 学習モデル」(Valiant) にある (PAC モデルについては，例えば Kearns and Vazirani [Kearns 94b] を参照)．最初にブースティングの問題，即ち PAC 学習モデルにおいてランダム予測より少し良い予測のできるいわゆる「弱学習アルゴリズム」を，任意の高い予測精度を持ついわゆる「強学習アルゴリズム」に変換することができるかという問題を提案したのは，Kearns and Valiant [Kearns 88, Kearns 94a] であった．1989 年，Schapire [Schapire 90] によって理論的に保証された最初のブースティング方式が与えられた．そして一年後，Freund [Freund 95] がより効率的である意味で最適なブースティング方式を考案したが，この方式も現実の使用には問題を抱えていた．これ

らの初期のブースティング方式を用いた実験結果としては，Drucker, Schapire and Simard [Drucker 93b] による文字認識 (OCR) 問題への適用がある．

3. AdaBoost

1995年に Freund and Schapire [Freund 97] は，初期のブースティング方式の問題点を改良し，AdaBoost アルゴリズムを考案した．この解説文ではこのアルゴリズムを中心に話を進める．AdaBoost は図 1 に示すアルゴリズムであり，その基本動作は以下のようである．まず，入力として訓練データ $(x_1, y_1), \ldots, (x_m, y_m)$ を受け取る．ここで，各 x_i は一定の領域または事例空間 X に属しており，また，各 y_i は一定のラベル集合 Y に属しているとする．この解説文では，後に多値問題についての拡張についても言及するが，概ね $Y = \{-1, +1\}$ であると仮定する．そして，AdaBoost は，与えられた「弱学習アルゴリズム」または「下位学習アルゴリズム」を，それぞれ 1 回呼び出すラウンドを T 回繰り返す $(t = 1, \ldots, T)$．ここで鍵になるアイデアは，訓練データ上に定義された確率分布 (または重み) によるリサンプリングを用いるということである．ラウンド t におけるこの分布による事例 i 上の重みを $D_t(i)$ と書く．これらの重みは初期値として全て等しく設定されるが，各ラウンドにおいて，誤って予測された事例の重みが増やされ，弱学習アルゴリズムがより難しい事例に集中して学習するようになっていくのである．

弱学習アルゴリズムに課された使命は，確率分布 D_t に対して適した「弱仮説」$h_t : X \to \{-1, +1\}$ を見つけ，出力することである．ここで，弱仮説 h_t の良さは，D_t による誤り確率

$$\epsilon_t = \Pr_{i \sim D_t}[h_t(x_i) \neq y_i] = \sum_{i : h_t(x_i) \neq y_i} D_t(i).$$

により測られる．ここで，D_t は弱学習アルゴリズムを訓練するのに用いられた事例の分布であることに注意する．現実的には，弱学習アルゴリズムは，訓練事例に加えて陽に重み D_t を入力として与えられてもよいし，または可能であれば訓練データから D_t によりリサンプルされたデータを用いて学習させても良い．

冒頭で出てきた競馬の例に戻れば，事例 x_i はレースの記述 (例えば出馬する馬の名前，それらのオッズ，過去の戦績など) にあたり，ラベル y_i は，各々のレースの結果 (即ち勝ち馬) の指定にあたる．また弱仮説は，分布 D_t により与えられたレースの集合に対してエキスパートにより与えられる経験則である．

一度弱仮説 h_t が得られると，AdaBoost は図 1 に示されているようにパラメータ α_t を設定する．直観的に言えば，α_t は h_t に付された重要度を表している．ここで，もし $\epsilon_t \leq 1/2$ であれば $\alpha_t \geq 0$ であり，前者は一般性を失わずに仮定できる．また，ϵ_t が小さければ小さいほど，α_t は大きくなる．次に，確率分布 D_t

Given: $(x_1, y_1), \ldots, (x_m, y_m)$ where $x_i \in X$, $y_i \in Y = \{-1, +1\}$

Initialize $D_1(i) = 1/m$.

For $t = 1, \ldots, T$:

- Train weak learner using distribution D_t.
- Get weak hypothesis $h_t : X \to \{-1, +1\}$ with error
$$\epsilon_t = \Pr_{i \sim D_t}[h_t(x_i) \neq y_i].$$
- Choose $\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right)$.
- Update:
$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases}$$
$$= \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$
where Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution).

Output the final hypothesis:

$$H(x) = \text{sign} \left(\sum_{t=1}^{T} \alpha_t h_t(x) \right).$$

図 1 ブースティングアルゴリズム AdaBoost.

が，図 1 に示すような規則で更新される．この更新規則は，h_t により誤って分類された事例の重みを増やし，正しく分類された事例の重みを減らしているのである．このようにして，重みは難しい事例に集中していくのである．最終仮説 H は，こうして得られた T 個の弱仮説 h_t を α_t を重みとして用いた重み付き多数決として得られる．

Schapire and Singer [Schapire 98c] は，AdaBoost アルゴリズムとその解析が，実数値による予測や信頼度付きの予測を出力する弱仮説についても対処できるよう拡張している．即ち，各々の事例 x について弱仮説 h_t は予測値 $h_t(x) \in \mathbb{R}$ を出力，予測値のサイン (正負) が予測ラベル (-1 または $+1$) を，予測値の絶対値 $|h_t(x)|$ が予測の信頼度を表現するような場合である．この解説文では，二値 ($\{-1,+1\}$) の予測を行なう弱仮説の場合に限定する．

4. 訓練誤差の解析

AdaBoost のもっとも基本的な理論的な性質は，訓練データ上の誤差を減らす能力についてである．弱仮説 h_t の誤差 ϵ_t を $1/2-\gamma_t$ と書くことにしよう．各々の事例のラベルをランダムに予測する仮説は，(二値問題では) 予測誤差 $1/2$ であるはずなので，γ_t は h_t の予測がランダム予測よりどの程度良いかを測るものである．Freund and Schapire [Freund 97] は，AdaBoost によって得られる最終仮説の訓練誤差 (即ち訓練データの誤って予測される事例の比率) は，以下のように上限されることを証明した．

$$\prod_t \left[2\sqrt{\epsilon_t(1-\epsilon_t)}\right] = \prod_t \sqrt{1-4\gamma_t^2} \leq \exp\left(-2\sum_t \gamma_t^2\right). \quad (1)$$

従って，もし各々の弱仮説がランダム予測より少しでも良ければ，ある特定の $\gamma > 0$ について $\gamma_t \geq \gamma$ が成り立ち，訓練誤差は指数関数的に減少することになる．

これに似た性質は，初期のブースティング方式も保有していたが，この下限値 γ がブースティングを行う前に陽に与えられている必要があった．現実的にはそのような下限値を知ることは難しい．これに対して，AdaBoost は，個々の弱仮説の誤差に適応するので，"adaptive" の最初の 3 文字をとって AdaBoost と名付けられた．

式 (1) で与えられた誤差の上限を，以下に与える汎化誤差の上限と組み合わせることにより，AdaBoost が確かにブースティング方式であることの証明を与えることができる．即ち，AdaBoost を用いれば，任意の分布に対して常にランダムより良い予測をする「弱学習アルゴリズム」を，効率的に (十分のサイズの訓練データを与えられれば) 任意に低い予測誤差を持つ仮説を出力できる「強学習アルゴリズム」に変換することができるのである．

5. 汎 化 誤 差

Freund and Schapire [Freund 97] は，AdaBoost の最終仮説の汎化誤差を，最終仮説の訓練誤差，サンプル数 m，弱仮説空間の VC 次元 d，およびブースティングのラウンド数 T の関数として上限した．ここで，VC 次元とは仮説空間の複雑性を測る指標である (例えば Blumer et al. [Blumer 89] 参照)．具体的には，Freund and Schapire は Baum and Haussler [Baum 89] のテクニックを用いて，汎化誤差が高い確率で高々

$$\hat{\Pr}[H(x) \neq y] + \tilde{O}\left(\sqrt{\frac{Td}{m}}\right)$$

であることを示した．ここで，$\hat{\Pr}[\cdot]$ は訓練データ上の経験分布を表す．この上限によれば，大きなラウンド数ブースティングを行うと (即ち，T が大きいと)，過学習をする可能性を示唆している．確かに過学習が起きる時もあるが，ブースティングを用いた初期の実験において複数の研究者 [Breiman 98, Drucker 96, Quinlan 96] が，数千ラウンド走らせた場合にでも，実験的に過学習しないことが多いと報告している．これにとどまらず多くの場合，AdaBoost は訓練誤差が 0 になった後にも汎化誤差を改良しつづけるという，上記の汎化誤差の上限とは一見矛盾する驚くべきことが観測された．例えば，図 2 の左式は "letter" データベースに対して Quinlan [Quinlan 93] の決定木学習アルゴリズム C4.5 を下位学習アルゴリズムとする AdaBoost を用いた場合の訓練データとテストデータに対する学習曲線を示している．

これらの実験的な結果を説明すべく，Schapire et al. [Schapire 98a] は訓練データ上の 'マージン' という概念を用いて，AdaBoost の汎化誤差に関して上記の解析とは性質の異なる解析を行った (この解析は Bartlett [Bartlett 98] による理論的な結果をふまえたものである)．ここで，訓練事例 (x,y) に対するマージンは，以下のように定義される．

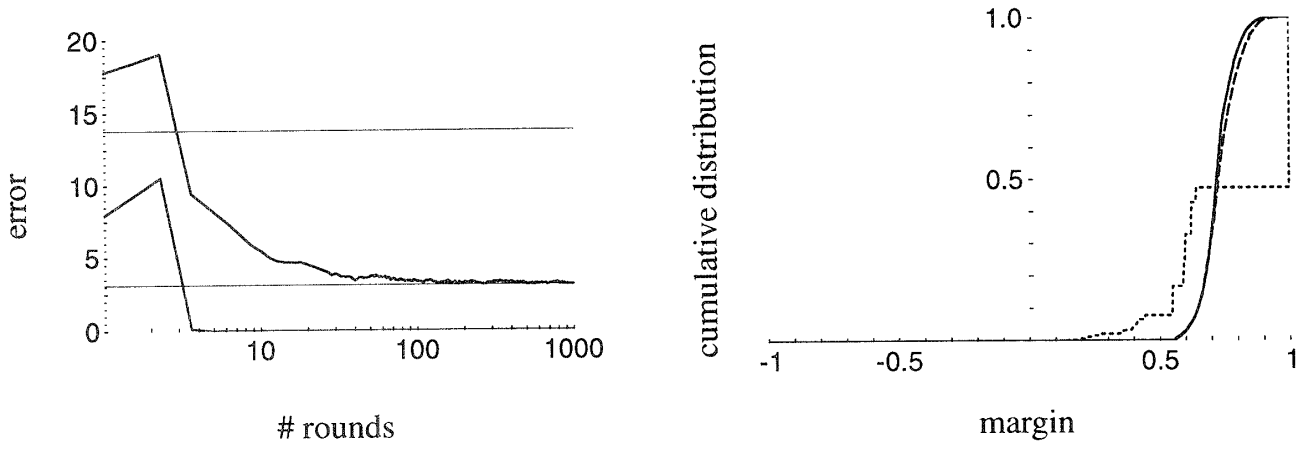


図2 'Letter' データセットに対する C4.5 を用いたブースティングによる学習精度 (誤り率) とマージン分布の推移 (Schapire et al. [Schapire 98a] による).
左：ブースティングのラウンド数の関数としての訓練データに対する予測誤り率 (下) とテストデータに対する予測誤り率 (上). 2 本の水平の線は各々下位学習アルゴリズムのテストデータに対する予測誤り率とブースティングにより得られた最終仮説のテストデータに対する予測誤り率を表す. 右：テストデータに対する予測マージンの累積分布 (点線：ブースティングラウンド 5 回後, ダッシュ線：100 回後, 実線：1000 回後).

$$\frac{y\sum_t \alpha_t h_t(x)}{\sum_t \alpha_t}. \qquad (2)$$

この量は, -1 と $+1$ の間の量であり, 仮説 H が事例 (x, y) を正しく分類する時に限って正の値をとる. さらに, マージンの絶対値は予測の信頼性を測るものと解釈できる. Schapire et al. は, より大きい訓練データ上のマージンが, より優れた汎化誤差の上限と対応することを示した. 具体的には, 任意の $\theta > 0$ に対して, 高い確率で汎化誤差が高々以下の量で上限されることを示した.

$$\hat{\Pr}\left[\text{margin}(x, y) \leq \theta\right] + \tilde{O}\left(\sqrt{\frac{d}{m\theta^2}}\right) \qquad (3)$$

ここで, この上限がブースティングのラウンド数 (T) に関して独立であることに注意したい. また, AdaBoost は, マージンの絶対値の小さい事例に集中するため, 訓練データのマージンの最大化に対して特に効果的であり, Schapire et al. はこれを定量的に示している. また, ブースティングのマージン最大化に関する効果は実験的にも観測されている. 例えば, 図 2 は "letter" データの訓練データ上のマージンの分布が, ブースティングのラウンドが進むにつれてどう変化するかをプロットしたものであるが, 訓練データ上の予測誤差が 0 になった後もブースティングはマージンを増やし続け, これに連動してテストデータに対する予測誤差が減っていることがわかる. 近年, 必ずしも成功しているとはいえないが, マージンの理論から示唆される知見を応用しようという試みが複数の研究者によってなされている [Breiman 97b, Grove 98, Mason 98].

AdaBoost の効果と挙動は, Freund and Schapire が示したように, ゲーム理論的な枠組みの中でも解釈することができる [Freund 96b, Freund 9?] (Grove and Schuurmans [Grove 98], Breiman [Breiman 97a] も参照). ブースティングは, ある種のゲームの繰り返しと解釈することが可能で, AdaBoost はこのような繰り返しゲームを行うより一般的なアルゴリズムの特別な場合であり, このゲームの近似解を見つける一つの方法であるということを示すことができる. このことはまた, ブースティングが線形計画法や逐次 (オンライン) 学習とも密接な関係があることを示している.

6. サポートベクトルマシーンとの関係

前章で述べたマージン理論による解釈は, ブースティングと (Vapnik 等による) サポートベクトルマシーン (SVM) との間に密接な関係があることを示唆している. この関係を明らかにするために, ここでは弱仮説は既に得られているとし, それらの間の重み付き多数決をとる際の重み α_t の選択の問題について考える. 考えられる一つの戦略は, 式 (3) の上限が最小化されるように係数を選ぶというものである. 特に, 第一項が零であるとし, 第二項の最小化に着目すると,

これは訓練データ上の最小のマージンの最大化と等価になる [Boser 92, Cortes 95, Vapnik 95][*1]．以下にこれについて詳しく説明する．まず，事例 (x, y) に対するすべての弱仮説の予測のベクトル（これを予測値ベクトルとよぶ）を $\mathbf{h}(x) \doteq \langle h_1(x), h_2(x), \ldots, h_N(x) \rangle$ と書き，係数のベクトル（これを重みベクトルとよぶ）を $\boldsymbol{\alpha} \doteq \langle \alpha_1, \alpha_2, \ldots, \alpha_N \rangle$ と書く．この記法と式 (2) で与えられたマージンの定義を用いて，最小マージンの最大化は以下のように書くことができる．

$$\max_{\boldsymbol{\alpha}} \min_{i} \frac{(\boldsymbol{\alpha} \cdot \mathbf{h}(x_i))y_i}{||\boldsymbol{\alpha}||\,||\mathbf{h}(x_i)||} \qquad (4)$$

ここで，ブースティングの場合，分母のノルムは以下のように定義される（h_t の値域がすべて $\{-1, +1\}$ である場合には，$||\mathbf{h}(x)||_\infty$ は常に 1 である）．

$$||\boldsymbol{\alpha}||_1 \doteq \sum_t |\alpha_t|, \quad ||\mathbf{h}(x)||_\infty \doteq \max_t |h_t(x)| .$$

これに対して，SVM の場合は式 (4) に示されているような最小マージンの最大化を陽な目標としている．ただし，ここでノルムとしてユークリッド距離を採用している．

$$||\boldsymbol{\alpha}||_2 \doteq \sqrt{\sum_t \alpha_t^2}, \quad ||\mathbf{h}(x)||_2 \doteq \sqrt{\sum_t h_t(x)^2}.$$

従って，SVM は予測ベクトルと重みベクトルの両方に ℓ_2 ノルム（ユークリッド距離）を用いており，AdaBoost は予測ベクトルには ℓ_∞ を用いて重みベクトルには ℓ_1 ノルムを用いているのである．このように記述すると SVM と AdaBoost は非常に似ているようであるが，いくつか重要な違いもある．

- **異なるノルムは異なるマージンに対応し得る．** ℓ_1 と ℓ_2 と ℓ_∞ の間の違いは，低次元の場合には特に注目すべきほどの違いにはならないかもしれない．しかし，ブースティングや SVM においては，次元は数百万単位になるなどきわめて高くなりがちである．このような場合には，ノルムの違いは非常に大きなマージンの値を生み得るのである．これは膨大な数の変数の中で関係のある変数の数が少ない時，即ち $\boldsymbol{\alpha}$ が疎である時に特に顕著となる．例えば，弱仮説の値域が $\{-1, +1\}$ であり，すべての事例のラベル y が k 個の弱仮説の多数決により計算できるとしよう．このような場合，関連ある弱仮説の数 k が弱仮説の総数に占める割合が小さければ，AdaBoost のマージンが SVM のそれよりもはるかに大きくなることを示すことができる．
- **必要な計算量が違う．** マージンの最大化に必要な計算は，複数の不等式を満たしながら目的関数の最大化を行ういわゆる数学的プログラミングの問題である．この意味からの両方式の違いは SVM は二次計画法に対応するが，AdaBoost は線形計画法に対応するということである（上でも記したように，AdaBoost と線形計画法との間には，ゲーム理論や逐次学習とも関連づける深い関係がある [Freund 96b]）．
- **高次元での探索を効率的に行うために異なるアプローチを用いている．** 二次計画法は線形計画法よりもより多くの計算量を要する．しかし，SVM とブースティング方式の間にははるかに重要な違いがある．SVM と AdaBoost が効果的であることの理由の一つは，極めて高次元の空間（場合によっては無限次元）において線形分離を行っていることである．すでに示したように過学習の問題はマージンの最大化によって避けられるが，高次元空間を扱うにあたっての計算量的な問題は残る．サポートベクトルマシンはこの計算量の問題に対してカーネル法を用いることにより，高次元「擬似」空間における内積と数学的に等価な低次元の計算を行うアルゴリズムを可能にしている．これに対してブースティングでは「欲張り探索」を用いている．この視点からは，弱学習アルゴリズムはラベル y と有意な相関を持つ $\mathbf{h}(x)$ の座標を求めるオラクルであると考えることができる．事例の重みの更新は，相関を測る事例上の分布の更新でもあるので，弱学習アルゴリズムに異なる相関を発見するように導いていることになる．SVM または AdaBoost を具体的な分類問題に適用するにあたっての労力のほとんどは，SVM の場合には適当なカーネル関数を，AdaBoost の場合は弱学習アルゴリズムを見つけることにある．カーネル関数と弱学習アルゴリズムは非常に違うので，両方式は多くの場合大きく異なる空間で探索を行い，得られる分類規則も大きく異なることが多いのである．

7. 多値問題への対処

これまでは，二つのクラスの分類を目的とした二値分類問題に限定して話を進めてきた．しかし，現実の

*1 AdaBoost は陽に最小マージンの最大化を行っているわけではないが，Schapire et al. [Schapire 98a] による解析によれば，AdaBoost はすべての訓練データのマージンを最大化しようとしていることが示唆されるので，この「最小マージン最大化アルゴリズム」を，概念的には AdaBoost の近似と考えることができる．実は，陽に最小マージンを最大化するアルゴリズムは，実験的に AdaBoost ほどに良い性能が得られていない [Breiman 97b, Grove 98].

問題の多く（もしくはほとんど）は，多値分類問題である．これまで AdaBoost を多値問題へ拡張する方式がいくつか提案されている．まず，もっとも単純な拡張である AdaBoost.M1 は，弱学習アルゴリズムが比較的精度が高く AdaBoost により構築される「難しい」分布に対してもある程度の精度が達成できる場合に有効な方式である．しかし，難しい分布に対する分類精度が 50%未満になってしまう場合には，この方式は有効でない．このような場合にも対処するため，より洗練された方式が開発されている．これらの多くは，多値分類問題をより規模の大きい二値問題に帰着することによりこの問題に対処している．例えば，Schapire and Singer [Schapire 98c] のアルゴリズム AdaBoost.MH は，すべての事例 x と可能なラベル y に対する「事例 x の正しいラベルは y であるか否か？」という形の質問を通して二値問題への帰着を行う．これに対して Freund and Schapire [Freund 97] のアルゴリズム AdaBoost.M2 (これは Schapire and Singer [Schapire 98c] の AdaBoost.MR アルゴリズムの特別な場合である) では，すべての事例 x，正しいラベル y と誤ったラベル y' に対する「事例 x の正しいラベルは y であるか y' であるか？」という質問を用いている．

これらの方式を用いるには，特別に弱学習アルゴリズムを設計しなくてはならないという欠点がある．これに対し，Dietterich and Bakiri [Dietterich 95] の誤り訂正のテクニックを利用した方法 [Schapire 97] では，通常の二値分類用の任意の弱学習アルゴリズムに対して用いることができ，しかも AdaBoost.MH や AdaBoost.M2 の誤差上限と似た上限を示すことが可能である．さらに，誤り訂正出力コードとブースティングを組み合わせるもう一つの方式が，Schapire and Singer [Schapire 98c] により考案されている．

8. 実験と応用

AdaBoost は現実に適用するにあたって多くの優れた性質を持っている．例えば，極めて単純でプログラムも簡単であり，計算効率的である．ラウンド数 T を別にすれば調整の必要なパラメータもない．また，弱学習アルゴリズムに対する何らの事前知識も必要としないので，弱仮説を発見するアルゴリズムであれば，任意のものと組み合わせることができる．しかも，弱学習アルゴリズムの予測精度と訓練データのサイズに対する緩やかな条件のもとで，理論的な性能保証が与えられている．これは，学習システム設計者にとっての大きな思考転換を意味する．即ち，学習領域全体において高精度を達成する学習アルゴリズムを苦労して発見しようとするかわりに，ランダム予測よりほんの少し良い精度を持つ弱学習アルゴリズムを発見すれば良いのである．

しかしながら，ブースティングも万能ではない．現実問題におけるブースティングの性能は，無論データと弱学習アルゴリズムの選択に依存する．理論の予測する通り，データ数が不足しているときや，複雑すぎる弱仮説が用いられたとき，または弱学習アルゴリズムが弱すぎるときには，ブースティングが効果的でない場合もある．また，ブースティングはデータ中の雑音に影響されやすいという報告もある [Dietterich 98](この点については後述する)．

AdaBoost は多くの研究者によって実験的に評価されている [Bauer 97, Dietterich 98, Drucker 96, Jackson 96, Maclin 97, Quinlan 96, Schwenk 98]．例えば Freund and Schapire [Freund 96a] は，UCI の機械学習ベンチマークデータ [Merz 98] を用いて，AdaBoost の性能評価を行っている．ここでは，C4.5 [Quinlan 93] 及び最良の単一ノードからなる決定木 (決定株) を見つける単純な弱学習アルゴリズムが用いられている．この実験結果の一部を図 3 に示す．この図からわかるように，決定株とブースティングを組み合わせるだけで，C4.5 と同等の精度が得られるし，また C4.5 と組み合わせることによって，C4.5 単体の性能を有意に改善できることが多い．

Schapire and Singer [Schapire 98b] による別の実験では，文書分類にブースティングを適用している．ここでは，単語またはフレーズの存在をテストする弱仮説が用いられている．この問題における AdaBoost と他の 4 方式との性能比較実験の結果の一部を図 4 に示す．ほとんどすべての場合において，ブースティングは比較対象のどの方式よりも有意に上回るか少なくとも同等の性能を達成していることがわかる．他にブースティングは，文書フィルタリング [Schapire 98d]，ランキング問題 [Freund 98]，および自然言語処理における分類問題 [Abney 99, Haruno 99] 等に応用されている．

Schapire and Singer [Schapire 98c] による AdaBoost の一般化は，ブースティングの最急降下法としての解釈を与えている．このアルゴリズムでは，各時点でのマージンに基づく「コスト」を割り振るポテンシャル関数が用いられている．このポテンシャル関数を用いることにより，AdaBoost の操作は弱仮説上の線形分離関数の空間での（座標毎の）最急降下法であると解釈することができる．この解釈を利用することにより，

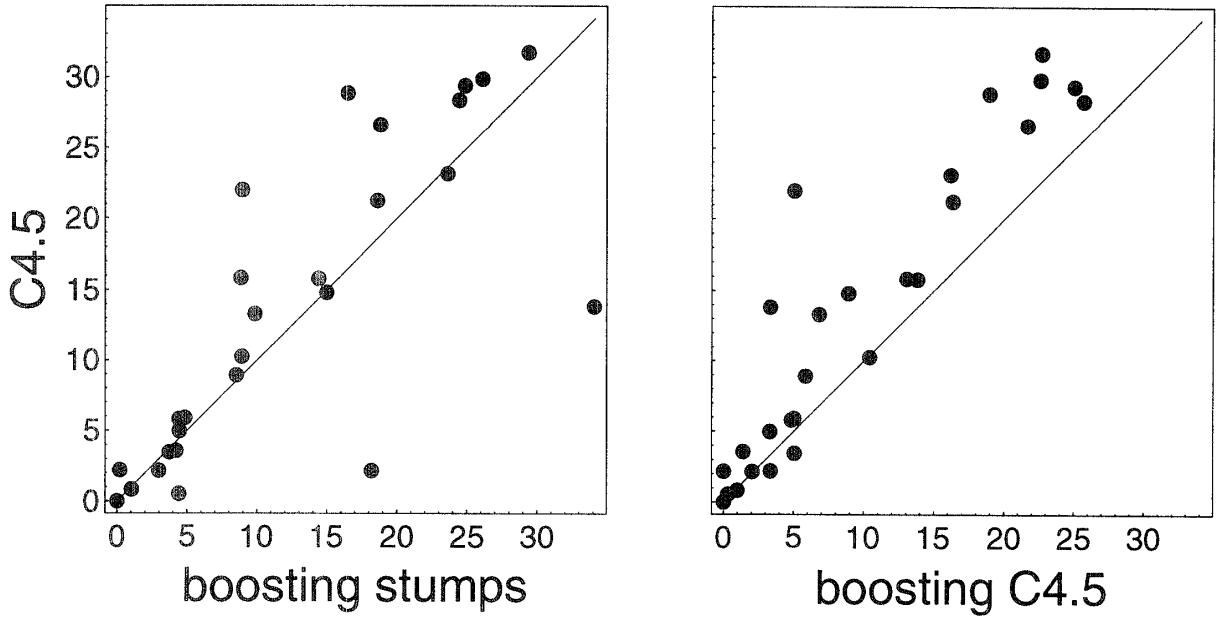


図 3　27 種のベンチマークデータを用いた C4.5，決定株+ブースティング，C4.5 +ブースティングの性能比較 (Freund and Schapire [Freund 96a] による).
各々の点は，二つの競合方式のテストデータに対する予測誤差の平均値を表す．y 軸の値は各々のグラフで C4.5 の平均予測誤差を表し，x 軸の値は左のグラフでは決定株+ブースティングの平均予測誤差を，右のグラフでは C4.5 +ブースティングの平均予測誤差を表す．すべての平均誤差は複数の試行回数の平均値である．

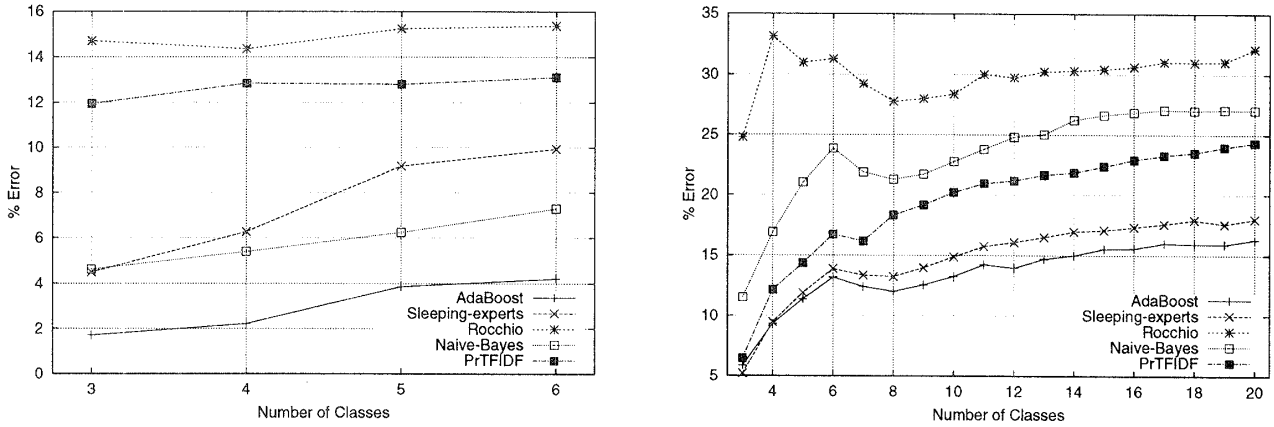


図 4　AdaBoost と 4 つのテキスト分類方式 (ナイーブベイズ方式，確率的 TF-IDF，'Rocchio'，'sleeping experts') の性能比較 (Schapire and Singer [Schapire 98b] による).
実験は二種類の文章データについて (ロイターニュース記事 (左) と AP ニュースヘッドライン (右))，そして多数の分類クラス数について行われた (x 軸がクラス数を表す).

一般的な分類規則の学習アルゴリズムの設計が可能となる．最近，Cohen and Singer [Cohen 99] は，ブースティングを RIPPER [Cohen 95] や IREP [Furnkranz 94] や C4.5rules [Quinlan 93] 等のシステムの生成するルールと類似のルールを学習する問題に適用している．また，Freund and Mason [Freund 99b] は，ブースティングを用いて決定木の一般化である alternating trees を学習する方法を提案している．

AdaBoost のもうひとつの興味深い適用法は，例外的な事例（いわゆる outlier）の発見，即ち訓練データの中で誤って分類されている事例や曖昧または分類不能な事例の発見をする方法としてである．AdaBoost は，もっとも困難な事例上に重みを集中させるので重みのもっとも大きな事例は，例外的な事例であることが多い．この現象に関して，Freund and Schapire [Freund 96a] による OCR の実験からとられた一例を図 5 に示す．例外的事例が非常に多いときは，困難な事例へ重みを集中することが，AdaBoost の性能に重大な悪影響を与えることがある．このことは，Dietterich [Dietterich 98] により非常に端的に示されている．この問題

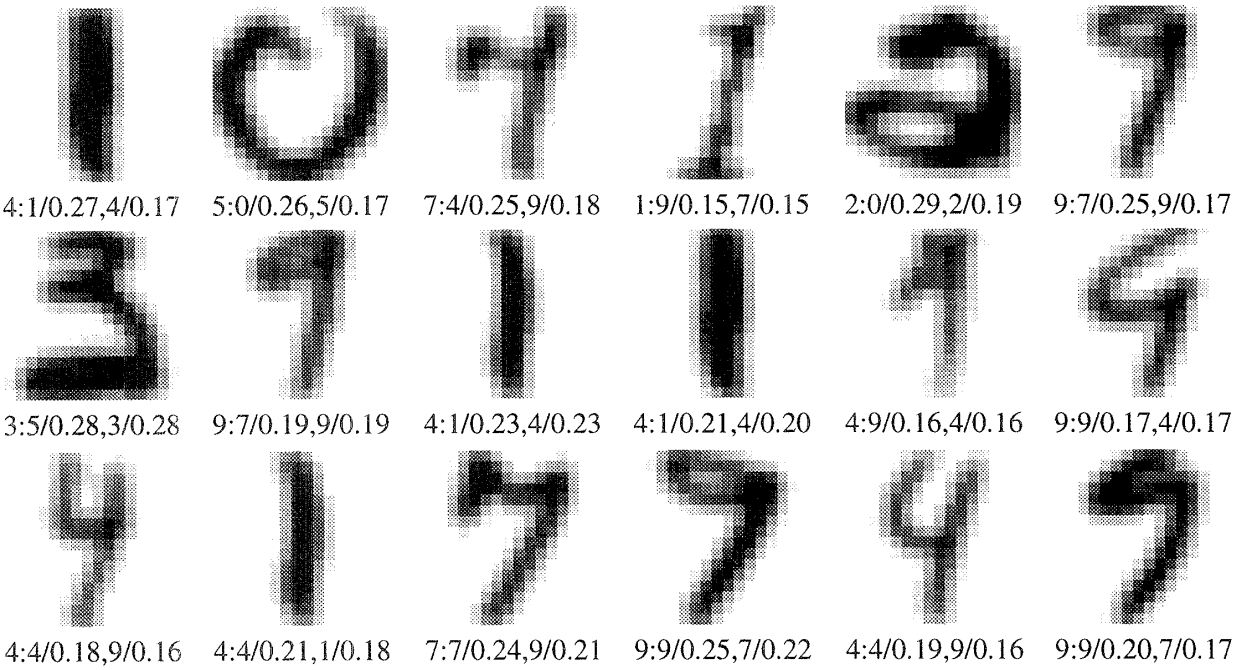


図5 OCRの実験において最も大きな重みを与えられた事例の例 (Freund and Schapire [Freund 96a] による). 上段の事例は，ブースティングを4ラウンド行った後のもの，中段は，12ラウンド後のもの，そして下段は，25ラウンド後のものである．各々の文字イメージの下の d：ℓ_1/w_1,ℓ_2/w_2 という形式の数値は，それぞれ d が事例のラベルを，ℓ_1 と ℓ_2 は各々その時点での仮説の投票において1番多くの重みを得たラベルと2番目に多くの重みを得たラベルを表す．w_1 と w_2 はこれらの正規化されたスコアを表す．

の解決のために，Friedman et al. [Friedman 98] は例外的な事例にあまり集中しない AdaBoost の変種 "Gentle AdaBoost" を提案している．また，最近 Freund [Freund 99a] は，この考え方をさらに推し進め，分類が難しすぎると判断された事例の重みは逆に減らすアルゴリズム "BrownBoost" を考案した．この方式は，Freund [Freund 95] の "boost-by-majority" アルゴリズムの適応型のものでる．この方式の解析と Schapire [Schapire 99] による "drifting games" に関する解析は，ブースティングとブラウン運動と繰り返しゲームとの間の興味深い関連を明らかにするとともに，多くの未解決問題と将来の研究課題を提示している．

◇参 考 文 献◇

[Abney 99] Steven Abney, Robert E. Schapire, and Yoram Singer. Boosting applied to tagging and PP attachment. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 1999.

[Bartlett 98] Peter L. Bartlett. The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network. *IEEE Transactions on Information Theory*, 44(2):525–536, March 1998.

[Bauer 97] Eric Bauer and Ron Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, to appear.

[Baum 89] Eric B. Baum and David Haussler. What size net gives valid generalization? *Neural Computation*, 1(1):151–160, 1989.

[Blumer 89] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the Association for Computing Machinery*, 36(4):929–965, October 1989.

[Boser 92] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*, pp. 144–152, 1992.

[Breiman 97a] Leo Breiman. Prediction games and arcing classifiers. Technical Report 504, Statistics Department, University of California at Berkeley, 1997.

[Breiman 97b] Leo Breiman. Arcing the edge. Technical Report 486, Statistics Department, University of California at Berkeley, 1997.

[Breiman 98] Leo Breiman. Arcing classifiers. *The Annals of Statistics*, 26(3):801–849, 1998.

[Cohen 95] William Cohen. Fast effective rule induction. In *Proceedings of the Twelfth International Conference on Machine Learning*, pp. 115–123, 1995.

[Cohen 99] William W. Cohen and Yoram Singer. A simple, fast, and effective rule learner. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, 1999.

[Cortes 95] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, September 1995.

[Dietterich 95] Thomas G. Dietterich and Ghulum Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Re-*

search, 2:263–286, January 1995.

[Dietterich 98] Thomas G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, to appear.

[Drucker 93b] Harris Drucker, Robert Schapire, and Patrice Simard. Boosting performance in neural networks. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(4):705–719, 1993.

[Drucker 96] Harris Drucker and Corinna Cortes. Boosting decision trees. In *Advances in Neural Information Processing Systems 8*, pp. 479–485, 1996.

[Freund 95] Yoav Freund. Boosting a weak learning algorithm by majority. *Information and Computation*, 121(2):256–285, 1995.

[Freund 96a] Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In *Machine Learning: Proceedings of the Thirteenth International Conference*, pp. 148–156, 1996.

[Freund 96b] Yoav Freund and Robert E. Schapire. Game theory, on-line prediction and boosting. In *Proceedings of the Ninth Annual Conference on Computational Learning Theory*, pp. 325–332, 1996.

[Freund 97] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, August 1997.

[Freund 98] Yoav Freund, Raj Iyer, Robert E. Schapire, and Yoram Singer. An efficient boosting algorithm for combining preferences. In *Machine Learning: Proceedings of the Fifteenth International Conference*, 1998.

[Freund 99a] Yoav Freund. An adaptive version of the boost by majority algorithm. In *Proceedings of the Twelfth Annual Conference on Computational Learning Theory*, 1999.

[Freund 99b] Yoav Freund and Llew Mason. The alternating decision tree learning algorithm. In *Machine Learning: Proceedings of the Sixteenth International Conference*, 1999.

[Freund 9?] Yoav Freund and Robert E. Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, to appear.

[Friedman 98] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Additive logistic regression: a statistical view of boosting. Technical Report, 1998.

[Furnkranz 94] Johannes Fürnkranz and Gerhard Widmer. Incremental reduced error pruning. In *Machine Learning: Proceedings of the Eleventh International Conference*, pp. 70–77, 1994.

[Grove 98] Adam J. Grove and Dale Schuurmans. Boosting in the limit: Maximizing the margin of learned ensembles. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, 1998.

[Haruno 99] Masahiko Haruno, Satoshi Shirai, and Yoshifumi Ooyama. Using decision trees to construct a practical parser. *Machine Learning*, 34:131–149, 1999.

[Jackson 96] Jeffrey C. Jackson and Mark W. Craven. Learning sparse perceptrons. In *Advances in Neural Information Processing Systems 8*, pp. 654–660, 1996.

[Kearns 88] Michael Kearns and Leslie G. Valiant. Learning Boolean formulae or finite automata is as hard as factoring. Technical Report TR-14-88, Harvard University Aiken Computation Laboratory, August 1988.

[Kearns 94a] Michael Kearns and Leslie G. Valiant. Cryptographic limitations on learning Boolean formulae and finite automata. *Journal of the Association for Computing Machinery*, 41(1):67–95, January 1994.

[Kearns 94b] Michael J. Kearns and Umesh V. Vazirani. *An Introduction to Computational Learning Theory*. MIT Press, 1994.

[Maclin 97] Richard Maclin and David Opitz. An empirical evaluation of bagging and boosting. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, pp. 546–551, 1997.

[Mason 98] Llew Mason, Peter Bartlett, and Jonathan Baxter. Direct optimization of margins improves generalization in combined classifiers. Technical report, Deparment of Systems Engineering, Australian National University, 1998.

[Merz 98] C. J. Merz and P. M. Murphy. UCI repository of machine learning databases, 1998. www.ics.uci.edu/~mlearn/MLRepository.html.

[Quinlan 93] J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[Quinlan 96] J. R. Quinlan. Bagging, boosting, and C4.5. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pp. 725–730, 1996.

[Schapire 90] Robert E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.

[Schapire 97] Robert E. Schapire. Using output codes to boost multiclass learning problems. In *Machine Learning: Proceedings of the Fourteenth International Conference*, pp. 313–321, 1997.

[Schapire 98a] Robert E. Schapire, Yoav Freund, Peter Bartlett, and Wee Sun Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26(5):1651–1686, October 1998.

[Schapire 98b] Robert E. Schapire and Yoram Singer. BoosTexter: A boosting-based system for text categorization. *Machine Learning*, to appear.

[Schapire 98c] Robert E. Schapire and Yoram Singer. Improved boosting algorithms using confidence-rated predictions. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, pp. 80–91, 1998. To appear, *Machine Learning*.

[Schapire 98d] Robert E. Schapire, Yoram Singer, and Amit Singhal. Boosting and Rocchio applied to text filtering. In *SIGIR '98: Proceedings of the 21st Annual International Conference on Research and Development in Information Retrieval*, 1998.

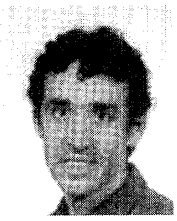
[Schapire 99] Robert E. Schapire. Drifting games. In *Proceedings of the Twelfth Annual Conference on Computational Learning Theory*, 1999.

[Schwenk 98] Holger Schwenk and Yoshua Bengio. Training methods for adaptive boosting of neural networks. In *Advances in Neural Information Processing Systems 10*, pp. 647–653, 1998.

[Valiant 84] L.G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, November 1984.

[Vapnik 95] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.

——著 者 紹 介——



Yoav Freund

ヨアブ・フロインドは，1982 年，数学と物理の学士号を Hebrew University より取得，その後 5 年間イスラエル軍において画像処理と実時間制御に携わった．1989 年には Eli Shamir の指導もと，Hebrew University より計算機科学の修士号を取得し，1993 年に Manfred Warmuth の指導のもと，U.C.Santa Cruz より計算機科学の博士号 (Ph.D.) を取得した．博士論文は "Data Filtering and Distribution Modeling Algorithms for Machine Learning" であった．その後，1993 年に AT&T Labs (元 AT&T Bell Labs) の研究員となり現在に至る．機械学習の理論と応用，および機械学習と情報理論，ゲーム理論，統計との関係に興味を持つ．〈yoav@research.att.com〉

Robert Schapire

ロバート・シャピリは，1986 年に数学と計算機科学の学士号を Brown University より取得，1991 年に Ronald Rivest の指導のもと MIT より修士，博士号 (Ph.D.) を取得した．"The Design and Analysis of Efficient Learning Algorithms" と題された博士論文は，1991 年の ACM Doctoral Dissertation Award を獲得した．Harvard 大学で短期間博士研究員を務めた後，AT&T Labs (元 AT&T Bell Labs) の研究員 (technical staff) となり現在に至る．機械学習の理論と応用，特にブースティングとオンライン学習に興味を持つ．〈schapire@research.att.com〉