LA-008

分割と併合に基づくブースティング
Boosting Based on Divide and Merge

竹内 寛明 † 柳下 英輝 ‡ 瀧本 英二 § 丸岡 章 §
Hiroaki Takeuchi Hideki Yagishita Eiji Takimoto Akira Maruoka

1. はじめに

ブースティングとは，精度の低い複数の仮説（弱仮説）を統合することによって精度の高い仮説（統合仮説）を作り出す手法である．最近，情報量の概念を用いて設計されたアルゴリズムがいくつか提案されている [1, 2, 3]．特に，InfoBoost は，各弱仮説 h_t（$1 \le t \le T$）の与える情報量が 0 でない限り，これらの弱仮説を統合することによって得られる仮説（統合仮説）の分類誤差が 0 に収束するという性質を持つ [1]．本稿ではこれを改良し，より効率よく情報を引き出すアルゴリズムの設計を試みる．まず，h_t によって新しく得られる情報の量を最大化することを目指すアルゴリズム M.InfoBoost を与える．M.InfoBoost は，$h_1, \ldots, h_t$ の取る値に基づいてサンプルの分割を行い，各領域ごとにパラメータを設定するように動作する．こうして得られた統合仮説は，t が小さいときは精度の向上が著しいが，t が大きくなるとむしろ精度が悪くなるという現象がしばしば観察される．これは，統合仮説が指数的に複雑になるために生じる「過学習」によるものと考えられる．そこで，領域数の指数的な増加を抑えるため，いくつかの領域を併合する過程を組み込んだアルゴリズム BP.InfoBoost を提案する．そして，理論的解析と実験的解析から，BP.InfoBoost の優位性を示す．

2. M.InfoBoost

X を事例空間，$Y = \{-1, +1\}$ をラベル空間とする．学習アルゴリズムは，サンプル $S = \{(x_1, y_1), \ldots, (x_m, y_m)\} \subseteq X \times Y$ が与えられると，その背後にある規則を推測して，これを仮説 $h : X \to Y$ として出力する過程である．ブースティングとは，性能のあまり良くない学習アルゴリズム（弱学習者）WL と，これを用いて精度の高い仮説を作ることを目指すブースターからなるスキームで，次のように動作する．各ラウンド $t (1 \le t \le T)$ において，(1) ブースターはサンプル S 上の確率分布 D_t を作り，これを S と共に弱学習者 WL に与える．(2)WL は弱仮説 h_t をブースターに返す（$h_t = \mathrm{WL}(S, D_t)$ と記す）．(3) ブースターは，分布を D_{t+1} に更新する．T ラウンド終了後，ブースターはこれまで得られた弱仮説 $h_1, \ldots, h_T$ を統合して仮説 F_T として出力する．

以下，エントロピー関数として $G(p) = 2\sqrt{p(1-p)}$ を用いる．各 t ラウンドにおいて，(X, Y) を確率 $D_t(i)$ で値 (x_i, y_i) を取る確率変数とみなす．分布 D_t の下で $Y = 1$ となる確率を $p = \Pr_{D_t}[Y = 1] = \sum_{i=1}^{m} D_t(i)(y_i + 1)/2$ とおくと，分布 D_t の下での Y のエントロピーは

$$H_{D_t}(Y) = G(p) = 2\sqrt{p(1-p)},$$

仮説 h_t を得たことによる Y の条件つきエントロピーは

$$H_{D_t}(Y|h_t) = \sum_{z \in \{-1,1\}} \Pr_{D_t}[h_t(X) = z] G(p_z)$$

と表される．ここに，$p_z = \Pr_{D_t}[Y = 1 \mid h_t(X) = z]$．これ以降, ある定数 $\gamma > 0$ が存在して，各弱仮説 h_t は

$$H_{D_t}(Y|h_t) \le (1 - \gamma) H_{D_t}(Y) \tag{1}$$

を満たすと仮定する．これは，WL が与えられた分布に対して何らかの情報を持つ仮説 h_t を返すことを保証するものである．

従来の InfoBoost の手法では，h_t が与えられると，h_t がラベル Y に関して無相関となるように分布を D_{t+1} に更新する．すなわち，InfoBoost において，新しい分布 D_{t+1} は

$$H_{D_{t+1}}(Y|h_t) = 1 \tag{2}$$

を満たす．仮定より，次の仮説 h_{t+1} は $H_{D_{t+1}}(Y|h_{t+1}) \le 1 - \gamma$ を満たすので，次の弱仮説 h_{t+1} は h_t にはない情報をもたらすことになる．

M.InfoBoost はこの考え方を発展させ，(2) の代わりに，これまでに得られたすべての弱仮説 $h_1, \ldots, h_t$ が無相関となるような分布，すなわち，

$$H_{D_{t+1}}(Y|h_1, \ldots, h_t) = 1 \tag{3}$$

を満たす D_{t+1} に分布を更新する．これにより，次の仮説 h_{t+1} は，これまでの弱仮説にない真に新しい情報をもたらすようになる．ただし，分布を更新するために，すべての弱仮説の値 $h_{1..t}(x_i) \equiv (h_1(x_i), \ldots, h_t(x_i))$ に基づいてサンプルを分割し，各 $\ell \in \{-1, +1\}^t$ に対して適当な実数 $\alpha[\ell]$ を定め，これを用いて領域 $\{(x_i, y_i) \mid h_{1..t}(x_i) = \ell\}$ ごとに分布の調整を行なう必要がある．この領域は，深さ t のすべての頂点に h_t がラベル付けされた完全 2 分決定木の葉に対応していることに注意されたい．よって，以下ではこの領域を葉 ℓ と呼ぶ．M.InfoBoost の詳細を図 1 に示す．統合仮説 F_T の性能に関して，次の定理が成り立つ．ただし，U はサンプル S 上の一様分布とする．

定理 1

$$\begin{aligned} \Pr_U[F_T(X) \ne Y] &\le H_U(Y|h_{1..T}) \\ &\le \prod_{t=1}^{T} H_{D_t}(Y|h_t) \le (1-\gamma)^T. \end{aligned}$$

これより，$T = O(1/\gamma \ln 1/\epsilon)$ ラウンドで F_T の訓練誤差が ϵ 以下となることが分かる．しかし, F_T の大きさは $O(2^T) = (1/\epsilon)^{O(1/\gamma)}$，すなわち，$1/\gamma$ に関して指数的になってしまう．

† 三菱ふそうトラック・バス（株）IT 本部
‡ NTT コムウェア（株）システム本部
§ 東北大学大学院情報科学研究科

Input：$S = \{(x_1, y_1), \ldots, (x_m, y_m)\} \subseteq X \times Y$
Initialize $D_1(i) = \frac{1}{m}$;
For $t = 1$ to T do
　$h_t = \mathrm{WL}(S, D_t)$;
　For $\ell \in \{-1, +1\}^t$ choose $\alpha_t[\ell] \in R$;
　Update D_t to D_{t+1} so that
　　$H_{D_{t+1}}(Y|h_{1..t}) = 1$;
Output: $F_T(x) = \mathrm{sign}\left(\sum_{t=1}^{T} \alpha_t[h_{1..t}(x)] h_t(x)\right)$;

図 1: M.InfoBoost

3. BP.InfoBoost

M.InfoBoost では，各ラウンドごとに葉の数が 2 倍ずつ増加してしまう．そこで，いくつかの葉を併合することにより葉の数を多項式程度に抑える過程を組み込んだ BP.InfoBoost を提案する．t ラウンド終了時の葉の集合を L_t とおく．L_t は $\{-1, +1\}^t$ の分割で表現される．すなわち，葉 $\ell \in L_t$ は $\{-1, +1\}^t$ の部分集合であり，領域 $\{(x_i, y_i) \mid h_{1..t}(x_i) \in \ell\}$ を表す．事例 x に対し，x が属する葉を関数 ℓ_t で表す．すなわち，

$$\ell_t(x) = \ell \Leftrightarrow h_{1\cdots t}(x) \in \ell.$$

一般に，葉 ℓ に到達する道は複数存在するので，関数 ℓ_t は決定木ではなくブランチングプログラムを表す．

BP.InfoBoost は，各ラウンド t において葉の分割と併合の 2 つのフェーズからなる．BP.InfoBoost の概要を図 2 に示す．分割のフェーズでは，新しい弱仮説 h_t の取る値に従って葉の数が 2 倍に増える．ここで，事例 x を新しい葉に対応させる関数を ℓ'_t とおくと，分割によるエントロピーの減少について

$$H_U(Y|\ell'_t) \le H_{D_t}(Y|h_t) H_U(Y|\ell_{t-1}) \le (1-\gamma) H_U(Y|\ell_{t-1})$$

が成り立つことを示すことができる.(実際，これを繰り返し用いると定理 1 が得られる.) 併合のフェーズでは，[2] の手法を用いることにより，葉の数を $O((1/\gamma)^2 (\ln 1/\epsilon)^2)$ に抑える．ℓ_t は，併合後のブランチングプログラムを表す関数となる．併合後のエントロピーの上昇について，$H_U(Y|\ell'_t) \ge \epsilon$ ならば，

$$H_U(Y|\ell_t) \le (1 + \gamma/2) H_U(Y|\ell'_t)$$

が成り立つ．こうして，各ラウンドごとに，エントロピーが $(1-\gamma)(1+\gamma/2) \le 1 - \gamma/2$ 倍以下に減少する．以上の議論より，次の定理を得る．

定理 2 BP.InfoBoost は $T = O(1/\gamma \ln 1/\epsilon)$ ラウンドで $\mathrm{Pr}_U(F_T(X) \ne Y) \le \epsilon$ を達成する．このときの統合仮説 F_T の大きさは，$O\left(\frac{1}{\gamma^3}\left(\ln \frac{1}{\epsilon}\right)^3\right)$ となる．

この定理より，BP.InfoBoost は，M.InfoBoost と同等の収束速度でエントロピーを減少させながら，M.InfoBoost に比べると複雑さが対数的に小さい統合仮説を生成することが分かる．

BP.InfoBoost の実際の性能を他のさまざまなブースティング手法と比較した実験の結果を図 3 に示す．実験

Input: $S = \{(x_1, y_1), \ldots, (x_m, y_m)\} \subseteq X \times Y$
Initialize $D_1(i) = \frac{1}{m}$, $L_0 = \emptyset$;
For $t = 1$ to T do
　$h_t = \mathrm{WL}(S, D_t)$
　[Divide]
　　Let $L' = \left\{\{(v, -1) | v \in \ell\} \mid \ell \in L_{t-1}\right\} \cup \left\{\{(v, +1) | v \in \ell\} \mid \ell \in L_{t-1}\right\}$;
　[Merge]
　　Partition L' into $L'_1, \ldots, L'_k$;
　　Let $L_t = \left\{\bigcup_{\ell \in L'_i} \ell \mid 1 \le i \le k\right\}$;
　For $\ell \in L_t$ choose $\alpha_t[\ell] \in R$;
　Update D_t to D_{t+1} so that $H_{D_{t+1}}(Y|\ell_t) = 1$;
Output: $F_T(x) = \mathrm{sign}\left(\sum_{t=1}^{T} \alpha_t[\ell_t(x)] h_t(x)\right)$;

図 2: BP.InfoBoost

では，弱学習者 WL として，与えられた分布 D_t の下でエントロピー $H_{D_t}(Y|h_t)$ が最小となる決定切り株（高さ 1 の決定木）h_t を返すものを用いた．UCI Repositry の車の好みに関するデータベースから，ランダムに抽出した 25%のデータをサンプル S として用い，得られた統合仮説 F_T の性能を残りの 75%のデータを用いて評価した．図より，M.InfoBoost が過適合を起こす様子が観察できる．一方，BP.InfoBoost は早いラウンドでの精度の向上が速いという M.InfoBoost の長所を生かしながら，過適合を起こすことなく精度の高い仮説を作り出すことが分かる．

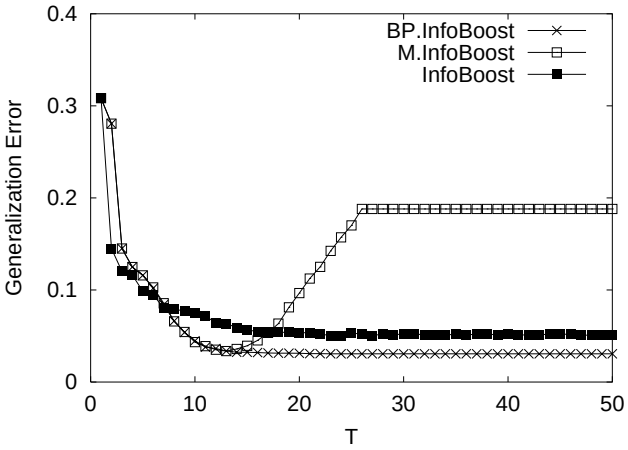

図 3: さまざまなブースティング手法の汎化誤差

参考文献

[1] J. Aslam. Improving algorithms for boosting. *13th COLT*, 200–207, 2000.

[2] Y. Mansour and D. McAllester. Boosting using branching programs. *13th COLT*, 220–224, 2000.

[3] E. Takimoto and A. Maruoka. Top-down decision tree learning as information based boosting. *Theoretical Computer Science*, 292(2):447–464, 2003.