Boostingと数理計画

（株）数理システム
山下浩
2003年9月

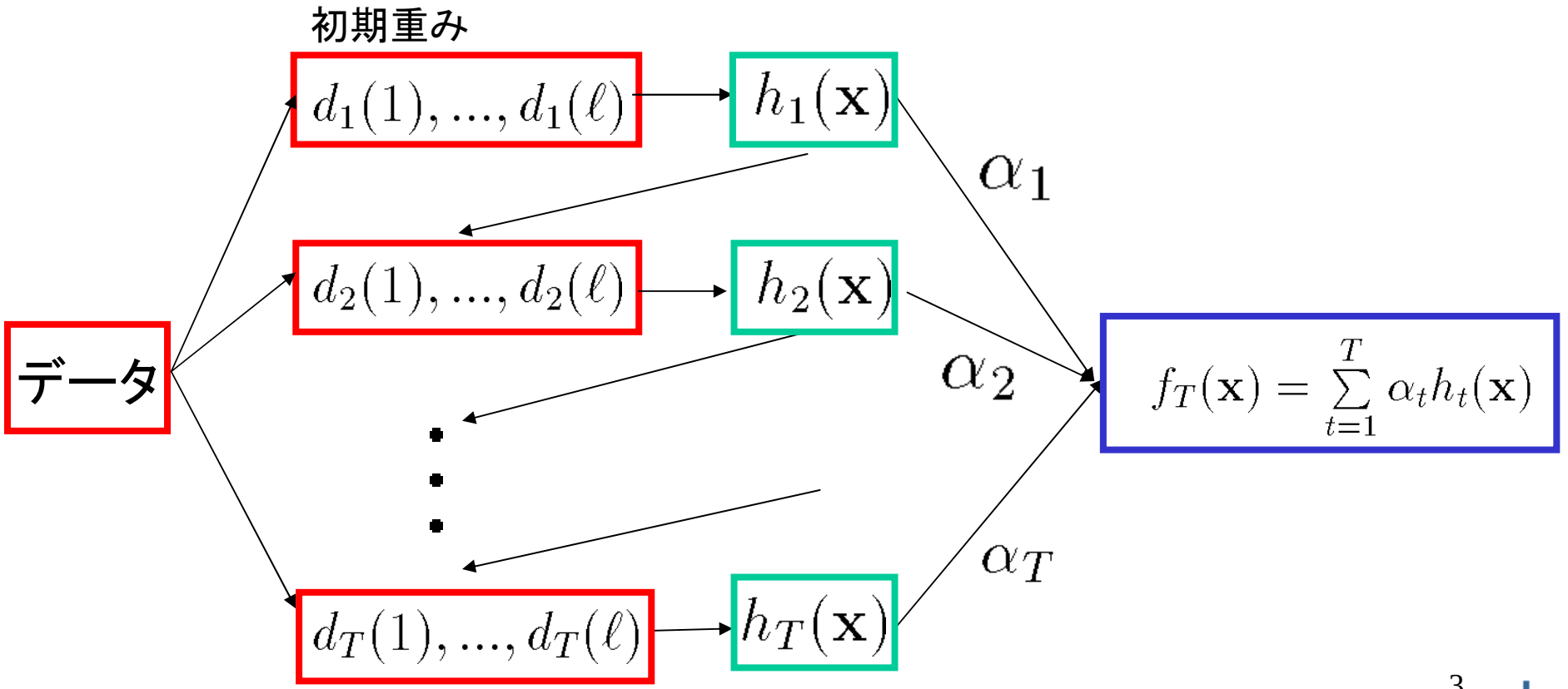
Boosting

- 弱学習機（weak learner---random guessより少し良い程度の能力）から性能の良い学習機を作り上げる.
- 基本的アイディアは，学習データの重みを変えて何回も学習する．前回に誤識別を起こしたデータの重みを大きくして再学習をする．これを繰り返すと，性能の良い学習機が可能になる.
- 初期の学習機（識別関数）は学習データ全体を対象とするが，後期の学習機は識別の困難なデータに特化した学習機となる.

Boostingの学習過程

- 学習用データ

$$S = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_\ell, y_\ell)\}, \quad \mathbf{x}_i \in \mathbb{R}^N, y_i \in \{-1, 1\}$$



AdaBoost (Freund and Schapire)

- 初期設定： $d_1(i) = 1/\ell, i = 1, ..., \ell$
- for $t = 1, ..., T$

(a) 重み付きデータセット $\{S, d\}$ より，仮説 $h_t : \mathbb{R}^N \to \{-1, 1\}$ を得る.

(b) 重み付きトレーニング誤差: $\varepsilon_t = \sum_{i=1}^{\ell} d_t(i) \frac{|h_t(\mathbf{x}_i) - y_i|}{2} (< 0.5)$

(c) $\alpha_t = \frac{1}{2} \log \frac{1-\varepsilon_t}{\varepsilon_t} (> 0)$

(d) 重みの更新：

$$d_{t+1}(i) = d_t(i) \exp\left\{-\alpha_t y_i h_t(\mathbf{x}_i)\right\} / Z_t$$

ここで，Z_t は $\sum_{i=1}^{\ell} d_{t+1}(i) = 1$ とするために規格化定数.

- 出力： $f(\mathbf{x}) = \mathrm{sgn}\left[f_T(\mathbf{x})\right], f_T(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})$

(d) 重みの更新：

$$
\begin{aligned}
d_{t+1}(i) &= d_t(i)\exp\left\{-\alpha_t y_i h_t(\mathbf{x}_i)\right\}/Z_t \\
&= \begin{cases} d_t(i)\exp\left\{-\alpha_t\right\}/Z_t & \text{correct :重みが減少} \\ d_t(i)\exp\left\{+\alpha_t\right\}/Z_t & \text{error :重みが増加} \end{cases}
\end{aligned}
$$

・トレーニング誤差：

$$\begin{aligned}\varepsilon_t &= \sum_{i=1}^{\ell} d_t(i)\frac{|h_t(\mathbf{x}_i)-y_i|}{2} = \sum_{error} d_t(i) \\ &= \sum_{error} d_{t-1}(i)\exp\left\{-\alpha_{t-1}y_i h_{t-1}(\mathbf{x}_i)\right\}/Z_{t-1} \\ &= \frac{1}{\ell}\sum_{error}\exp\left\{-y_i\sum_{r=1}^{t-1}\alpha_r h_r(\mathbf{x}_i)\right\}/\prod_{r=1}^{t-1}Z_r\end{aligned}$$

$$1-\varepsilon_t = \sum_{correct} d_t(i) = \frac{1}{\ell}\sum_{correct}\exp\left\{-y_i\sum_{r=1}^{t-1}\alpha_r h_r(\mathbf{x}_i)\right\}/\prod_{r=1}^{t-1}Z_r$$

AdaBoostの仕組み

・データ i の識別エラー: $\Longleftrightarrow y_i f(\mathbf{x}_i) = -1 \Longrightarrow \exp\{-y_i f_T(\mathbf{x}_i)\} > 1$

・データ i の正しい識別 : $\Longleftrightarrow y_i f(\mathbf{x}_i) = 1 \Longrightarrow \exp\{-y_i f_T(\mathbf{x}_i)\} < 1$

なので，損失関数：

$$\sum_{i=1}^{\ell} U(y_i f_T(\mathbf{x}_i)) = \sum_{i=1}^{\ell} \exp\{-y_i f_T(\mathbf{x}_i)\}$$

をなるべく小さくするように α_t を選ぶ.

$$\sum_{i=1}^{\ell} \exp\{-y_i(\alpha_t h_t(\mathbf{x}_i) + f_{t-1}(\mathbf{x}_i))\} \text{ の最小化}$$

$$\begin{aligned}
0 &= \frac{\partial}{\partial \alpha_t} \sum_{i=1}^{\ell} \exp\{-y_i(\alpha_t h_t(\mathbf{x}_i) + f_{t-1}(\mathbf{x}_i))\} \\
&= -\sum_{i=1}^{\ell} y_i h_t(\mathbf{x}_i) \exp\{-y_i(\alpha_t h_t(\mathbf{x}_i) + f_{t-1}(\mathbf{x}_i))\} \\
&= -\sum_{correct} e^{-\alpha_t - y_i f_{t-1}(\mathbf{x}_i)} + \sum_{error} e^{\alpha_t - y_i f_{t-1}(\mathbf{x}_i)} \\
&= e^{\alpha_t} \left\{ -e^{-2\alpha_t} \sum_{correct} e^{-y_i f_{t-1}(\mathbf{x}_i)} + \sum_{error} e^{-y_i f_{t-1}(\mathbf{x}_i)} \right\} \Longrightarrow \\
\alpha_t &= \frac{1}{2} \log \frac{\sum_{correct} e^{-y_i f_{t-1}(\mathbf{x}_i)}}{\sum_{error} e^{-y_i f_{t-1}(\mathbf{x}_i)}} = \frac{1}{2} \log \frac{1-\varepsilon_t}{\varepsilon_t}
\end{aligned}$$

トレーニング誤差の減少

$$
\begin{aligned}
error &= \frac{1}{\ell}\sum_{i=1}^{\ell}\frac{1}{2}\left|f(\mathbf{x}_i)-y_i\right| = \frac{1}{\ell}\sum_{i,y_i f(\mathbf{x}_i)<0} 1 \\
&\leq \frac{1}{\ell}\sum_{i=1}^{\ell}\exp\left\{-y_i\sum_{t=1}^{T}\alpha_t h_t(\mathbf{x}_i)\right\} = \prod_{t=1}^{T} Z_t
\end{aligned}
$$

$$
\begin{aligned}
Z_t &= \sum_{error} d_t(i)e^{\alpha_t} + \sum_{correct} d_t(i)e^{-\alpha_t} = \varepsilon_t e^{\alpha_t} + (1-\varepsilon_t)e^{-\alpha_t} \\
&= \varepsilon_t\sqrt{\frac{(1-\varepsilon_t)}{\varepsilon_t}} + (1-\varepsilon_t)\sqrt{\frac{\varepsilon_t}{1-\varepsilon_t}} = 2\sqrt{\varepsilon_t(1-\varepsilon_t)}
\end{aligned}
$$

- $\varepsilon_t = \frac{1}{2} - \gamma_t \leq \frac{1}{2} - \gamma$ ならば

$$\prod_{t=1}^{T} Z_t = \prod_{t=1}^{T}\sqrt{1-4\gamma_t^2} \leq e^{-2\sum\gamma_t^2} \leq e^{-2T\gamma^2} \to 0$$

いくつかの損失関数:

- AdaBoost: $U(z) = \exp\{-z\}$
- LogitBoost: $U(z) = \log(1 + \exp\{-2z\})$
- MadaBoost: $U(z) = \begin{cases} \frac{1}{2}\exp\{-2z\}, & z \geq 0 \\ -z + \frac{1}{2}, & z < 0 \end{cases}$

•AdaBoostはノイズ(outlier)の影響を受けやすい.
•しかし，一般にboostingの汎化能力が高いことは経験的に示されている. ⇒以下を参照。

・重み付き誤差

$$\varepsilon(h, \mathbf{d}) = \sum_{i=1}^{\ell} d_i \frac{1}{2} \left| h(\mathbf{x}_i) - y_i \right|, \quad \sum_{i=1}^{\ell} d_i = 1, d_i \geq 0, h(\mathbf{x}_i) \in \{-1, 1\}$$

・edge

$$\gamma(h, \mathbf{d}) = \sum_{i=1}^{\ell} d_i y_i h(\mathbf{x}_i), \quad \sum_{i=1}^{\ell} d_i = 1, d_i \geq 0, h(\mathbf{x}_i) \in \mathbb{R}$$

$$(\gamma(h, \mathbf{d}) = 1 - 2\varepsilon(h, \mathbf{d}), \text{ for } h(\mathbf{x}_i) \in \{-1, 1\})$$

・識別関数 f のマージン：$\rho(f) = y_i f(\mathbf{x}_i)$

Min-max定理（edge最小化＝マージン最大化）

$H = \{h_j \,|j = 1, ..., J\}$：仮説の組

$$\sum_{i=1}^{\ell} d_i = 1, \mathbf{d} \geq 0, \quad \sum_{j=1}^{J} w_j = 1, \mathbf{w} \geq 0$$

$$\gamma^* = \min_{\mathbf{d}} \max_{h_j \in H} \left\{ \sum_{i=1}^{\ell} d_i y_i h_j(\mathbf{x}_i) \right\} = \max_{\mathbf{w}} \min_{1 \leq i \leq \ell} y_i \left\{ \sum_{j=1}^{J} w_j h_j(\mathbf{x}_i) \right\} = \rho^*$$

$$\begin{array}{ll} \min & \gamma \\ \text{s.t.} & \sum_{i=1}^{\ell} d_i y_i h_j(\mathbf{x}_i) \leq \gamma, j = 1, ..., J \\ & \sum_{i=1}^{\ell} d_i = 1, \mathbf{d} \geq 0 \end{array}$$

$$\begin{array}{ll} \max & \rho \\ \text{s.t.} & \sum_{j=1}^{J} y_i w_j h_j(\mathbf{x}_i) \geq \rho, i = 1, ..., \ell \\ & \sum_{j=1}^{J} w_j = 1, \mathbf{w} \geq 0 \end{array}$$

マージン最大化を表わすLP問題
(LP boosting)

AdaBoostの識別関数

$$f_t(\mathbf{x}) = \sum_{r=1}^{t} \alpha_r h_r(\mathbf{x})$$

$$w_j^t = \sum_{r=1}^{t} \begin{cases} \alpha_r, & h_r = h_j \\ 0, & \text{otherwise} \end{cases}, \quad j = 1, ..., J$$

$$\Downarrow$$

$$\sum_{r=1}^{t} \alpha_r h_r(\mathbf{x}) = \sum_{j=1}^{J} w_j^t h_j(\mathbf{x})$$

$$\|\mathbf{w}\|_1 = \|\boldsymbol{\alpha}\|_1$$

・入力空間から特徴空間（有限次元）へ

$$\mathbf{x} \mapsto \boldsymbol{\phi}(\mathbf{x}) = (\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), ...)$$

・特徴空間での線形分離とマージン最大化

$$\rho = \max_{\mathbf{w}} \min_{i} \frac{y_i \langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x}_i) \rangle}{\|\mathbf{w}\|}$$

・特徴空間での数理計画問題：

$$\begin{aligned} \max \quad & \rho \\ \text{s.t.} \quad & y_i \langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x}_i) \rangle \geq \rho, i = 1, ..., \ell \\ & \|\mathbf{w}\| = 1 \end{aligned}$$

● ℓ_1 ノルムの場合 (boosting)

・特徴空間を仮説の集合とする：

$$\boldsymbol{\phi}(\mathbf{x}) = (h_1(\mathbf{x}), h_2(\mathbf{x}), ...)$$

・マージン：

$$\rho_i = \frac{y_i \langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x}_i) \rangle}{\|\mathbf{w}\|_1} = \frac{y_i \sum_{j=1}^{J} w_j h_j(\mathbf{x}_i)}{\sum_{j=1}^{J} w_j} = \frac{y_i \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x}_i)}{\sum_{t=1}^{T} \alpha_t}$$

($\mathbf{w} \geq 0, \boldsymbol{\alpha} \geq 0$ と仮定できる.)

● ℓ_2 ノルムの場合 (Support Vector Machine)

Boostingとペナルティ関数法

$$\begin{aligned} \max \quad & \rho \\ \text{s.t.} \quad & \frac{y_i \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x}_i)}{\sum_{t=1}^{T} \alpha_t} \geq \rho, i = 1, \ldots, \ell \\ & \boldsymbol{\alpha} \geq 0 \end{aligned}$$

・指数ペナルティ関数：

$$F(\rho, \boldsymbol{\alpha}) = -\boldsymbol{\rho} + \boldsymbol{\beta} \exp\left\{ \frac{1}{\boldsymbol{\beta}} \left(\rho - \frac{y_i \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x}_i)}{\sum_{t=1}^{T} \alpha_t} \right) \right\}, \beta > 0$$

⇩

Boostingはマージン最大化問題を近似的に解いているとみなせる.

⇒汎化能力

参考文献

- http://www.boosting.org/
- R. Meir and G. Rätsch. An introduction to boosting and leveraging. In S. Mendelson and A. Smola, editors, *Advanced Lectures on Machine Learning*, LNCS, pp 119-184. Springer, 2003.
- A.Demiriz, K.P. Bennet, and J.S-Taylor, Linear programming boosting via column generation, working paper, 2000.