数理解析研究所講究録 1439 巻 2005 年 111-127

ブースティングとそのロバスト化

統計数理研究所　竹之内 高志 (Takashi Takenouchi)
Department of Fundamental Statistical Theory, The Institute of Statistical Mathematics.
共同研究者:金森敬文, 村田昇, 江口真透

概要

判別問題において AdaBoost はシンプルかつ強力な方法であり, そのアルゴリズムは指数ロスを逐次最小化していると見なせる. AdaBoost は例題に対する重みを指数的に更新することで学習を行なうが, 重みの更新が急過ぎるが故に例題中に含まれる外れ値に影響されやすいという特徴を持つ. 例題に含まれる外れ値には入力, ラベルに対する外れ値があるが, 本稿では各種の外れ値に対してロバストなブースティングアルゴリズムを提案する.

Keywords: 判別分析, η-divergence, contamination モデル, 影響関数, most B-robust

1　導入

本稿ではブースティングを用いた 2 値の判別問題を扱う (Mclachlan, 1992). ブースティングとは精度の低い学習機, 弱学習機を組み合わせて精度の高い学習機を構成する手法である (Schapire, 1990). 一般にブースティングのアルゴリズムは弱学習機を用いたロス関数の逐次最小化によって得られ, 用いるロスによって得られる統計的性質が異なる. ブースティングの典型的な例がアダブースト (Freund and Schapire, 1997, Schapire, 1999) であり, アダブーストから派生した様々な亜種が存在する. ブースティングのアルゴリズムの主な特徴は与えられた例題に対する重みの分布を学習の各ステップで適応的に変える事であり, 対応したロス関数を逐次最小化することで得られる. アダブーストのアルゴリズムは指数ロス

$$L_{\exp}(F) = \sum_{i=1}^{N} \exp(-F(\boldsymbol{x}_i)y_i) \tag{1}$$

を逐次的に最小化することで得られる. ただし $\boldsymbol{x} \in \mathbf{R}^p$ を入力を表す特徴ベクトル, y をクラスラベルを表す変数とし, 例題として $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ が得られているとする. アダブーストは統計的にはロジスティック判別を逐次的に行っていると解釈する事ができ, またアダブーストと正値測度空間における最尤推定との幾何学的関係が情報幾何の立場から与えられている (Amari and Nagaoka, 2000, Lebanon and Lafferty, 2001). アダブーストは高速かつ精度よく判別を行うことができるが, 一方で外れ値に対してロバストではないという欠点も併せ持っており, アルゴリズムのロバスト化のために様々な改良が成されている (Rätsch et al., 2001, Takenouchi and Eguchi, 2004). 本稿

ではアダブーストの欠点を克服するために外れ値に対してロバストなブースティングアルゴリズム，すなわち，ロバストなロス関数に関して考察する．

判別問題において例題に含まれる外れ値は大きく 2 種類に分類することが出来る．ひとつは $\boldsymbol{x}$ の空間における外れ値であり，もうひとつは y における外れ値である．この二つの大きな違いは $\boldsymbol{x}$ は一般に連続値を取りうる一方で y は ± 1 にしか値を取らないため外れ値がミスラベル $(y \to -y)$ として解釈される事にある (Copas, 1988)．したがって例題にどちらの外れ値が含まれているかによって用いるべきアプローチは異なるべきである．外れ値に対してロバストな手法を考える時にその基準となる

本稿ではミスラベルを考慮した確率モデルに対応したブースティングアルゴリズム，係数の推定の立場から影響関数を用いたアルゴリズムを導出する．

2 アダブースト

ブースティングに用いる弱学習機を $f(\boldsymbol{x}) \in \{1, -1\}$, 弱学習機の集合を $\mathcal{F}$ とする．また定義関数を

$$\mathrm{I}(R) = \begin{cases} 1, & \mathrm{R} \text{ が真}, \\ 0, & \text{その他}, \end{cases}$$

としてアダブーストのアルゴリズムは以下の様に書ける．

1. 重みの初期値を $w_1(i) = \frac{1}{N}$, 判別関数を $F_0(\boldsymbol{x}) = 0$ とする．
2. For $t = 1, \cdots, T$
   (a) 重みつき誤り率

$$\varepsilon_t(f) = \sum_{i=1}^{N} w_t(i)\mathrm{I}(f(\boldsymbol{x}_i) \neq y_i),$$

   を最小にする弱学習機を選択する．

$$f_t = \operatorname*{argmin}_{f \in \mathcal{F}} \varepsilon_t(f).$$

   (b) 選択した弱学習機に対する係数

$$\alpha_t = \frac{1}{2} \log \frac{1 - \varepsilon_t(f_t)}{\varepsilon_t(f_t)},$$

   を計算し学習機を以下で更新する．

$$F_t = F_{t-1} + \alpha_t f_t.$$

   (c) 重みを以下のように更新する．

$$w_{t+1}(i) = \frac{\exp(-F_t(\boldsymbol{x}_i)y_i)}{Z_{t+1}},$$

   ただし Z_{t+1} は正規化定数で $Z_{t+1} = \sum_{i=1}^{N} \exp(-F_t(\boldsymbol{x}_i)y_i)$ とする．

3. 判別関数 $F_T(\boldsymbol{x}) = \sum_{t=1}^{T} \alpha_t f_t(\boldsymbol{x})$ の符合で判別.

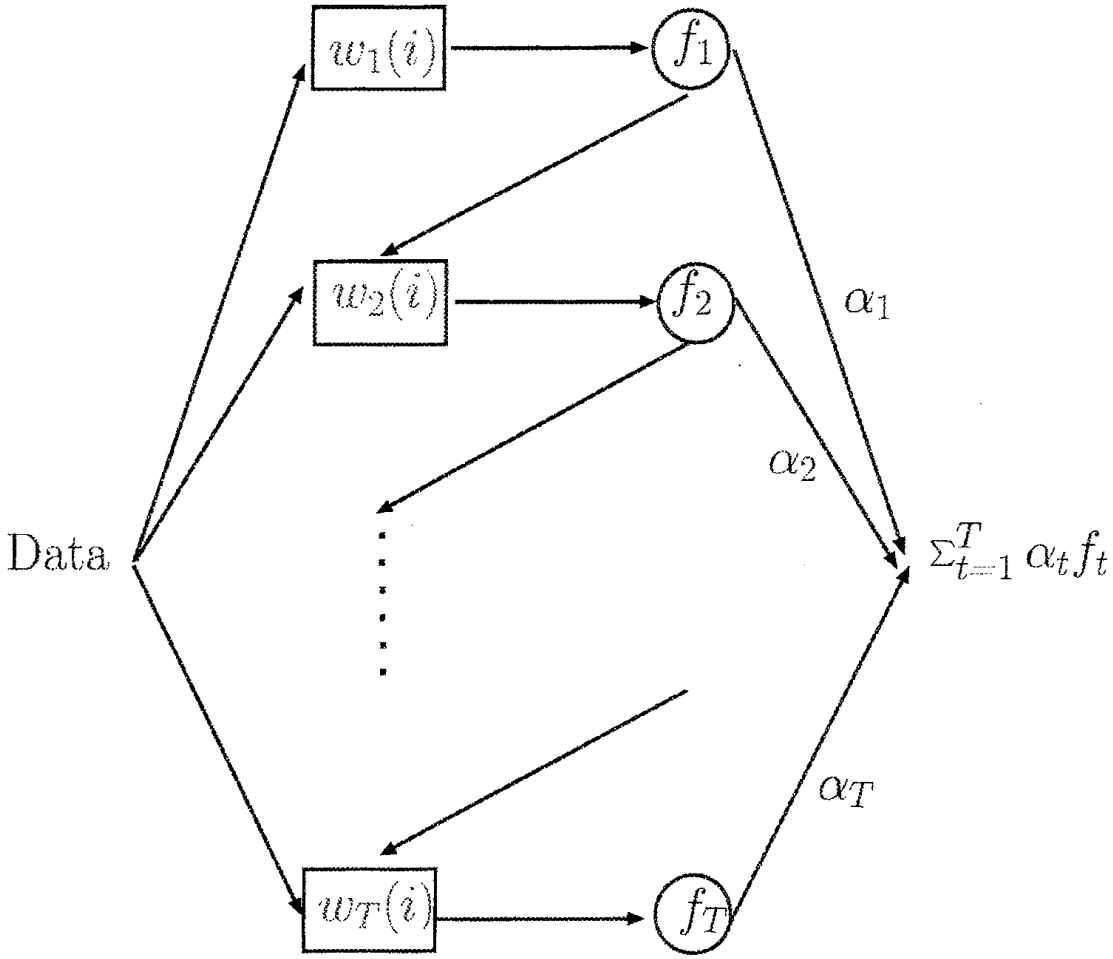

図 1: アダブーストのアルゴリズムの流れ.

図 (1) はアルゴリズムの流れを示した物であり，逐次的に更新される重みを用いて学習が進んでいく．上記のアルゴリズムの特徴をいくつか述べておく．2.(a) において選択される weak learner $f_t(\boldsymbol{x})$ は $\varepsilon_t(f_t) \leq \frac{1}{2}$ を満たしている（仮に $\varepsilon_t(f_t) > \frac{1}{2}$ であるならば $-f_t(\boldsymbol{x})$ を用いればよい）．また 2.(b) において計算する α_t は $\varepsilon_t(f_t)$ の log-odds であり，最終的な判別機における $f_t(\boldsymbol{x})$ の信頼度を表す量である．$\varepsilon_t(f_t)$ が小さければ α_t は大きく，逆に $\varepsilon_t(f_t)$ が $\frac{1}{2}$ に近い場合には α_t は小さくなる (図 2)．

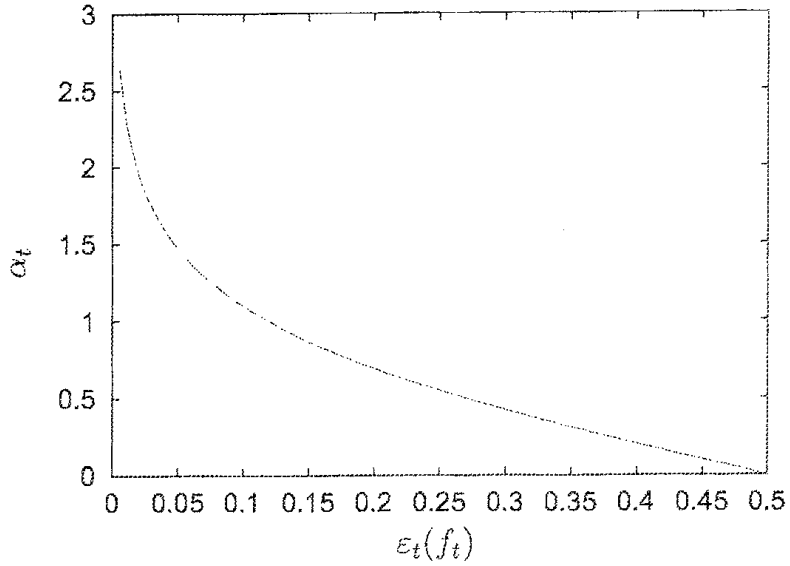

図 2: 重みつき誤り率に対する学習機の係数 α_t のグラフ．

2.(c) の重みの更新については以下のような事が言える．

$$w_{t+1}(i) \propto \begin{cases} w_t(i)e^{\alpha_t} & \text{if } f_t(\boldsymbol{x}_i) \neq y_i \\ w_t(i)e^{-\alpha_t} & \text{その他} \end{cases} . \tag{2}$$

つまり $f_t(\boldsymbol{x})$ が間違った例題の重みは e^{α_t} 倍し，正解した場合には重みを $e^{-\alpha_t}$ 倍する．加えてこの更新則は

$$\varepsilon_{t+1}(f_t) = \frac{1}{2} \quad (t = 1, \cdots, T-1). \tag{3}$$

という性質を持っている．つまり $(t+1)$ ステップの重み $w_{t+1}(i)$ の下では $f_t(\boldsymbol{x})$ は最弱の学習機となっている．言い換えれば判別機 $f_t(\boldsymbol{x})$ が最も苦手な重みで次のステップの学習を行い判別機を選択するということである．

AdaBoost はシンプルで強力な方法であるが，外れ値を含む様な例題に対してロバストでないという事が知られている．AdaBoost がなぜロバストでないのかを簡単な例を用いて説明する．例えば，1 個の線形な判別機によって全ての例題が完全に判別できる場合を考える．この例題中の 1 つの例題に人為的にミスラベルを起こしたとする．この様な状況の下で AdaBoost を適用す

るとアルゴリズムはラベルが間違った例題に対する重みを指数的に増加させるので, 2.(a) においてラベルの間違った例題に特化した weak learners を選択する様になり, 結果として, 最終的に出力される判別機もラベルの間違った例題に大きく影響を受け, テストデータに対する性能が減少してしまう.

通常外れ値に対するロバストネスは幾何的な解釈に基づいており, 外れ値とは特徴空間において例題の塊から離れた点であると定義される. そしてある手法が外れ値によって影響を受けにくい場合に, その手法はロバストであるという. 一方, 今我々は2値の判別問題について考えているのでラベルの空間は $\{1,-1\}$ のみであり, $\boldsymbol{x}$ における外れ値と y における外れ値では取るべきアプローチを変えるべきである. 次章では外れ値は 1 と -1 の間の置換, つまり y が $-y$ に置換されることによって起こっていると考え, 確率的なアプローチ, すなわちラベル y の分布 $p(y|\boldsymbol{x})$ が確率 η で

$$(1-\eta)p(y|\boldsymbol{x})+\eta p(-y|\boldsymbol{x})$$

の様に汚染を受けていると仮定する (Copas, 1988). ミスラベルの確率 η は $\boldsymbol{x}$ に依存してもよい. 上記の確率モデルに対応したブースティングアルゴリズムとして η-ブーストを導出する. アルゴリズムとしては AdaBoost に簡単な修正を加えたものになっており, その特徴は, 一様な重みを考慮したアルゴリズムになっている事である. η-Boost は外れ値に影響されづらく, ロバストな判別機を構成する事ができる.

3 η-ブースト

3.1 アルゴリズム

AdaBoost に簡単な修正を加えて以下の η-Boost を得る. ここでは $0 \leq \eta < 1$ とする.

1. 重みの初期値を $w_1^*(i)=\dfrac{1}{N}\ (i=1,\cdots,N)$ とし, $F_0(\boldsymbol{x})=0$ とする.
2. for $t=1,\cdots,T$
   (a) 重みつき誤り率を最小にする学習機を選択する.
   $$f_t(\boldsymbol{x})=\mathrm{argmin}_{f\in\mathcal{F}}\varepsilon_t^*(f).$$
   ただし $\varepsilon_t^*(f)=\sum_{i=1}^N w_t^*(i)\mathrm{I}(f(\boldsymbol{x}_i)\neq y_i)$.
   (b) 選択した学習機の係数
   $$\alpha_t^*=\log\frac{\sqrt{1-\varepsilon_t(f_t)+(\eta K_t)^2}+\eta K_t}{\sqrt{\varepsilon_t(f_t)}}$$
   を計算し, 判別関数を $F_t=F_{t-1}+\alpha_t^* f_t$ で更新する. ただし $\varepsilon_t(f)$ は AdaBoost で定義された誤り率, $K_t=\dfrac{(1-2\varepsilon_1(f_t))}{2\sqrt{\varepsilon_t(f_t)}}\left(\dfrac{(1-\eta)Z_t}{N}\right)^{-1}$ とする.

(c) $w^*_{t+1}(i) = \dfrac{(1-\eta)e^{-F_t(\boldsymbol{x}_i)y_i} + \eta}{Z^*_{t+1}}$ で重みを更新する．ただし $Z^*_{t+1} = \sum_{i=1}^{N}(1-\eta)e^{-F_t(\boldsymbol{x}_i)y_i} + \eta$.

3. 判別関数 $\mathrm{sgn}(\sum_{t=1}^{T}\alpha^*_t f_t(\boldsymbol{x}))$ を出力し符号で判別を行う．

$\eta = 0$ とおけば上のアルゴリズムは AdaBoost となる．2.(b) で計算する α^*_t は最終的な判別関数 $\sum_{t=1}^{T}\alpha^*_t f_t(\boldsymbol{x})$ における weak learner $f_t(\boldsymbol{x})$ の信頼度とみなせる．$\varepsilon_t(f_t)$ は AdaBoost で定義された重み $w_t(i)$ で評価した $f_t(\boldsymbol{x})$ の重みつき誤り率，$\varepsilon_1(f_t)$ は一様な重み $w_1(i) = \frac{1}{N}$ で評価した誤り率である．α^*_t に関して，η-Boost は $\varepsilon^*_t(f_t)$ の log-odds を ηK_t によって調節している．仮に $\varepsilon_1(f_t) \geq \frac{1}{2}$，つまり一様な重み $w_1(i)$ の下で $f_t(\boldsymbol{x})$ の性能が低い場合には $K_t \leq 0$ となり α^*_t は log-odds を縮小した値となる．逆に $\varepsilon_1(f_t) \leq \frac{1}{2}$，つまり $f_t(\boldsymbol{x})$ が $w_1(i)$ の下でもある程度意味のあるものであれば α^*_t は log-odds よりも大きな値をとる．2.(c) の重みの更新則は以下の様に書く事ができる．

$$w^*_{t+1}(i) = (1-\delta^*_{t+1})w_{t+1}(i) + \delta^*_{t+1}w_1(i), \tag{4}$$

$$\delta^*_{t+1} = \frac{\eta N}{(1-\eta)Z_{t+1} + \eta N}. \tag{5}$$

つまり $w^*_{t+1}(i)$ は指数的に更新される AdaBoost の重み $w_{t+1}(i)$ を一様な重み $w_1(i)$ で緩和していると解釈する事ができる．η-Boost は一様な重みを考慮した (4) を使う事によって，例題中のノイズに特化した weak learner を選びにくくしている．また η-Boost においても $\varepsilon^*_{t+1}(f_t) = \frac{1}{2}$ が成立している事に注意する．

3.2 アルゴリズムの導出

η-ブーストのアルゴリズムは関数 $U_\eta(z) = (1-\eta)e^z + \eta z$ から生成されるロス関数

$$L_\eta(F) = \sum_{i=1}^{N} U_\eta(-F(\boldsymbol{x}_i)y_i) \tag{6}$$

の逐次最小化によって得ることが出来る．ナイーブエラーロス関数を

$$L_{naive}(F) = \sum_{i=1}^{N} -F(\boldsymbol{x}_i)y_i$$

と定義すると，(6) は以下の様に書く事ができる．

$$L_\eta(F) = (1-\eta)L_{\exp}(F) + \eta L_{naive}(F).$$

つまり η-ブーストのロス関数はアダブーストの指数ロスとナイーブエラーロスをパラメーター η でつないだロスとなっている．

今 $F_0(\boldsymbol{x}_i) = 0 (i = 1, \cdots, N)$ から出発し, $F_t(\boldsymbol{x}) = \sum_{s=1}^{t} \alpha_s^* f_s(\boldsymbol{x})$ が構成できたとする. この時 $F_t(\boldsymbol{x})$ に適当な $\alpha f(\boldsymbol{x})$ を加える事によってロス関数 $L_\eta(F_t + \alpha f)$ を最小化したい. まず $f(\boldsymbol{x})$ に関して最適化を行なう.

$$L(F_t + \alpha f) \geq L(F_t) + \left. \frac{\partial L(F_t + \alpha f)}{\partial \alpha} \right|_{\alpha=0} \alpha \tag{7}$$

より $L(F_t + \alpha f)$ の代わりに

$$\left. \frac{\partial L(F_t + \alpha f)}{\partial \alpha} \right|_{\alpha=0} = \sum_{i=1}^{N} -U'_\eta(-F_t(\boldsymbol{x}_i) y_i) f(\boldsymbol{x}_i) y_i$$

を $f(\boldsymbol{x})$ に関して最小化する. $U'_\eta(F_t(\boldsymbol{x}_i) y_i) \propto w_{t+1}^*(i)$ であるから上記の最適化は η-ブーストの 2.(a) の $\varepsilon_{t+1}^*(f)$ の最小化と等価であり, その解を $f_{t+1}(\boldsymbol{x})$ とする. α に関しては以下を陽に解く事ができる.

$$\alpha_{t+1}^* = \mathrm{argmin}_\alpha L(F_t + \alpha f_{t+1}) \tag{8}$$

この解が η-ブーストのアルゴリズムにおける α_{t+1}^* と等しくなる. 上記の操作を順次繰り返す事により η-ブーストのアルゴリズムが得られる.

3.3 ロス関数の性質

本章ではロス関数を生成する関数 $U_\eta(z)$ の性質について考察する (Friedman et al., 2000). アブストラクトなロスを

$$\mathrm{E}\left[(1-\eta) e^{-F(\boldsymbol{X}) Y} - \eta F(\boldsymbol{X}) Y\right] \tag{9}$$

とし, これを最小化する判別関数 $F(\boldsymbol{x})$ を求める. ただし E は $(\boldsymbol{X}, Y)$ の確率密度関数 $p(\boldsymbol{x}, y)$ に関する期待値とする. $\boldsymbol{x}$ で条件付けて最小化を行なえば十分である (Friedman et al., 2000).

$$\frac{\partial \mathrm{E}[(1-\eta) e^{-F(\boldsymbol{x}) Y} - \eta F(\boldsymbol{x}) Y | \boldsymbol{X} = \boldsymbol{x}]}{\partial F(\boldsymbol{x})} = 0 \tag{10}$$

とすると以下の式を得る.

$$\log \frac{p(1|\boldsymbol{x})}{p(-1|\boldsymbol{x})} = \log \frac{(1-\eta) e^{F_\eta^*(\boldsymbol{x})} + \eta}{(1-\eta) e^{-F_\eta^*(\boldsymbol{x})} + \eta}. \tag{11}$$

これは以下と等価である.

$$p(y|\boldsymbol{x}) = \frac{(1-\eta) e^{F_\eta^*(\boldsymbol{x}) y} + \eta}{(1-\eta)(e^{F_\eta^*(\boldsymbol{x})} + e^{-F_\eta^*(\boldsymbol{x})}) + 2\eta}. \tag{12}$$

ただし $F_\eta^*(\boldsymbol{x})$ は以下で定義する.

$$F_\eta^*(\boldsymbol{x}) = \mathrm{argmin}_F \mathrm{E}[(1-\eta) e^{-F(\boldsymbol{x}) Y} - \eta F(\boldsymbol{x}) Y | \boldsymbol{X} = \boldsymbol{x}].$$

$$\log \frac{p(1|\boldsymbol{x})}{p(-1|\boldsymbol{x})} = 0 \Leftrightarrow F_\eta^*(\boldsymbol{x}) = 0 \tag{13}$$

となりベイズ境界と $F_\eta^*(\boldsymbol{x})$ で定まる境界は等しい事がわかる. 次章では $F_\eta^*(\boldsymbol{x})$ の境界以外での性質について考察する.

3.4 ミスラベルモデル

本章では学習ステップ数 T, weak learners のベクトル $\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \cdots, f_T(\boldsymbol{x}))'$, パラメーター $\boldsymbol{\alpha_0} \in \mathbf{R}^T$ を固定する. $p_{\eta_0}(\boldsymbol{x}, y)$ を $(\boldsymbol{X}, Y)$ の確率密度関数とし, 以下を仮定する.

$$p_{\eta_0}(\boldsymbol{x}, y) = p(\boldsymbol{x}) p_{\eta_0}(y|\boldsymbol{x}).$$
$$p_{\eta_0}(y|\boldsymbol{x}) = \frac{(1-\eta_0)e^{\boldsymbol{\alpha}_0 \cdot \boldsymbol{f}(\boldsymbol{x})y} + \eta_0}{(1-\eta_0)(e^{\boldsymbol{\alpha}_0 \cdot \boldsymbol{f}(\boldsymbol{x})} + e^{-\boldsymbol{\alpha}_0 \cdot \boldsymbol{f}(\boldsymbol{x})}) + 2\eta_0}.$$

ただし $0 \leq \eta_0 < 1$, $\boldsymbol{f}(\boldsymbol{x})$ と $\boldsymbol{\alpha}$ の内積を $\boldsymbol{\alpha} \cdot \boldsymbol{f}(\boldsymbol{x})$ と記す. これは (12) で $\eta = \eta_0$, $F^*_{\eta}(\boldsymbol{x}) = \boldsymbol{\alpha}_0 \cdot \boldsymbol{f}(\boldsymbol{x})$ とした場合である. 一般に η_0 は未知であるから, $\boldsymbol{\alpha}$ の最適化は $U_{\eta_1}(z)$ から生成したロス関数を用いる. アブストラクトなロス関数を同時最適化をした時の解を $\boldsymbol{\alpha}(\eta_1)$ と書く. すなわち

$$\boldsymbol{\alpha}(\eta_1) = \mathrm{argmin}_{\boldsymbol{\alpha}} \mathrm{E}_{\eta_0} \left[U_{\eta_1}(-\boldsymbol{\alpha} \cdot \boldsymbol{f}(\boldsymbol{X})Y) \right]. \tag{14}$$

ここで E_{η_0} は $p_{\eta_0}(\boldsymbol{x}, y)$ に関する期待値とする. $\boldsymbol{\alpha}(\eta_1)$ は以下を満たす.

$$\mathrm{E}_{\eta_0} \left[\boldsymbol{f}(\boldsymbol{X}) Y U'_{\eta_1}(-\boldsymbol{\alpha}(\eta_1) \cdot \boldsymbol{f}(\boldsymbol{X})Y) \right] = 0. \tag{15}$$

$\boldsymbol{\alpha}(\eta_1)$ から構成される判別関数 $\mathrm{sgn}(\boldsymbol{\alpha}(\eta_1) \cdot \boldsymbol{f}(\boldsymbol{x}))$ は η_1 に依存する. 今判別関数 $\boldsymbol{\alpha}(\eta_1) \cdot \boldsymbol{f}(\boldsymbol{x})$ のアブストラクトなエラーレイト

$$\mathrm{Err}(\boldsymbol{\alpha}(\eta_1) \cdot \boldsymbol{f}) = \mathrm{E}_{\eta_0} \left[\mathrm{I}(\boldsymbol{\alpha}(\eta_1) \cdot \boldsymbol{f}(\boldsymbol{X})Y < 0) \right] \tag{16}$$

を考え, (14) の下で最もエラーレイトの低い η_1 を求める. この時以下の定理を得る.

定理 1: 全ての $0 \leq \eta_1 < 1$ に対して

$$\mathrm{Err}(\boldsymbol{\alpha}(\eta_1) \cdot \boldsymbol{f}) \geq \mathrm{Err}(\boldsymbol{\alpha}(\eta_0) \cdot \boldsymbol{f}) = \mathrm{Err}(\boldsymbol{\alpha}_0 \cdot \boldsymbol{f}). \tag{17}$$

証明 まず $\boldsymbol{\alpha}(\eta_0) = \boldsymbol{\alpha}_0$ を証明する. $\eta_1 = \eta_0$ とすると $\boldsymbol{\alpha}_0$ は (15) を満たす事から $\boldsymbol{\alpha}_0$ は (14) の解の一つである. またロス関数 $U_{\eta_1}(z)$ は凸であることから (14) の解がただ一つである事がわかり, $\boldsymbol{\alpha}(\eta_0) = \boldsymbol{\alpha}_0$ が言える. ところでベイズルールを以下のように記す.

$$\lambda_{\eta_0}(\boldsymbol{x}) = \log \frac{p_{\eta_0}(1|\boldsymbol{x})}{p_{\eta_0}(-1|\boldsymbol{x})}. \tag{18}$$

ベイズルールはアブストラクトなエラーレイト (16) を最小化し (Mclachlan, 1992), $\boldsymbol{\alpha}(\eta_0) \cdot \boldsymbol{f}(\boldsymbol{x}) > 0 \Leftrightarrow \lambda_{\eta_0}(\boldsymbol{x}) > 0$ が成立することから以下が示せる.

$$\mathrm{Err}(\boldsymbol{\alpha}(\eta_0) \cdot \boldsymbol{f}) = \mathrm{Err}(\lambda_{\eta_0}). \tag{19}$$

以上より全ての $0 \leq \eta_1 < 1$ に対して (17) が成立する.

この定理は例題が (14) から生成している場合にロス関数 $U_{\eta_0}(z)$ を用いた時, つまり η-Boost において $\eta = \eta_0$ とした時にエラーレイト (16) が最も小さくなる事を示している. 実際には 10-フォールド クロスバリデーションを用いて η_0 を推定する事ができる.

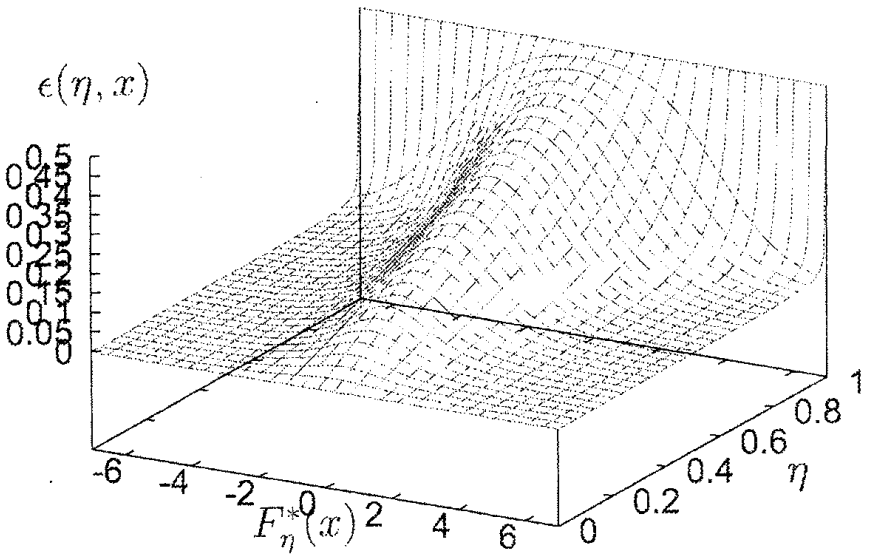

図 3: $\epsilon(\eta_0, z)$ のグラフ.

3.5 モデルの解釈

本章ではモデル (14) が持つ統計的な意味について考察する. 今 $p_{\eta_0}(y|\boldsymbol{x})$ がロジスティックモデル

$$p_0(y|\boldsymbol{x}) = \frac{e^{\boldsymbol{\alpha}_0 \cdot \boldsymbol{f}(\boldsymbol{x})y}}{e^{\boldsymbol{\alpha}_0 \cdot \boldsymbol{f}(\boldsymbol{x})} + e^{-\boldsymbol{\alpha}_0 \cdot \boldsymbol{f}(\boldsymbol{x})}} \tag{20}$$

からどの程度離れているかを考えると

$$p_{\eta_0}(y|\boldsymbol{x}) = (1 - \epsilon(\eta_0, \boldsymbol{\alpha}_0 \cdot \boldsymbol{f}(\boldsymbol{x})))\, p_0(y|\boldsymbol{x}) + \epsilon(\eta_0, \boldsymbol{\alpha}_0 \cdot \boldsymbol{f}(\boldsymbol{x}))p_0(-y|\boldsymbol{x}), \tag{21}$$

$$\epsilon(\eta_0, z) = \frac{\eta_0}{(1-\eta_0)(e^z + e^{-z}) + 2\eta_0} \tag{22}$$

となる. つまりモデル (14) は $\epsilon(\eta_0, \boldsymbol{\alpha}_0 \cdot \boldsymbol{f}(\boldsymbol{x}))$ という確率でロジスティックモデルが汚染されていると解釈することが出来る. ただし $\epsilon(\eta_0, \boldsymbol{\alpha}_0 \cdot \boldsymbol{f}(\boldsymbol{x}))$ は η_0, $\boldsymbol{\alpha}_0$, $\boldsymbol{x}$ の値に依存した量であり, $\boldsymbol{\alpha}_0 \cdot \boldsymbol{f}(\boldsymbol{x}) = 0$ の時に最大化され, その最大値は $\frac{\eta_0}{2}$ である. この事からミスラベルは境界の近くで最も頻繁に起こっており, 境界から離れるに従ってミスラベルが起こる確率は指数的に減少する事がわかる (図 3).

この境界付近にミスラベルが多く判別が困難であるという状況は現実の問題に対しても有効に働く (Takenouchi and Eguchi, 2004).

4 影響関数によるロバスト化

本章では $\boldsymbol{x}$ の外れ値に対してロバストなアルゴリズムに対応したロス関数を導出する (Kanamori et al.,2004). 外れ値による影響を弱学習機の係数の推定 α の立場から考察し, 外れ値が存在する下でのロバストネスの指標としてグロスエラーセンシティビティを用いる (Hampel et al., 1986). グロスエラーセンシティビティを最小にする推定量はモスト B-ロバストであると言われる. 一般にはモスト B-ロバストな推定量は有効な推定量ではない事に注意する. 本章ではある統計モデルが真の確率構造を含む場合に, そのモデルに関連するロスの中でモスト B-ロバストなロスを導出する.

4.1 ロス $U(z)$ によるアルゴリズム

今 $U(z)$ を凸な単調増加関数とし得られた例題と判別関数 $F(\boldsymbol{x}) = \sum_{t=1}^{T} \alpha_t f_t(\boldsymbol{x})$ に対してロス関数

$$L_U(F) = \sum_{i=1}^{N} U(-F(\boldsymbol{x}_i)y_i)$$

の逐次最小化によるブースティングアルゴリズムを考える (Eguchi and Copas, 2001, Mason et al.,1999, Murata et al.,2004).

1. 重みの初期値を $w_1(i) = \frac{1}{N}$,$F_0(\boldsymbol{x}) = 0$ とする.

2. For $t = 1, \cdots, T$

   (a) 重みつき誤り率 $\varepsilon_t(f) = \sum_{i=1}^{N} w_t(i)\mathrm{I}(f(\boldsymbol{x}_i) \neq y_i)$ を最小にする学習機を選択する.

   $$f_t = \operatorname*{argmin}_f \varepsilon_t(f),$$

   (b) 選択した学習機に対して係数

   $$\alpha_t = \operatorname*{argmin}_{\alpha} \frac{1}{N} \sum_{i=1}^{N} U(-y_i F_{t-1}(\boldsymbol{x}_i) - \alpha y_i f_t(\boldsymbol{x}_i))$$

   を計算し判別関数を更新する.

   $$F_t = F_{t-1} + \alpha_t f_t.$$

   (c) 重みを更新する.

   $$w_{t+1}(i) = \frac{U'(-F_t(\boldsymbol{x}_i)y_i)}{Z_{t+1}},$$

   ただし Z_{t+1} は正規化定数とする.

$F_T(\boldsymbol{x}) = \sum_{t=1}^{T} \alpha_t f_t(\boldsymbol{x})$ の符合で判別を行う.

重みの更新がロス $U(z)$ の導関数 $U'(z)$ によって定義されている点に注意する.

関数 $U(z)$ にから導出したアルゴリズムにどのような性質があるかを考察する. 本章では簡単のため $f_1(\boldsymbol{x}), \cdots, f_T(\boldsymbol{x})$ は既知とし $\boldsymbol{\alpha} = (\alpha_1, \cdots, \alpha_T)$ の推定のみに着目する. 今アブストラクトなロス $\mathrm{E}[U(-F(\boldsymbol{X})Y)]$ を考えると最小にする関数 $F^*(\boldsymbol{x})$ は

$$\frac{p(1|\boldsymbol{x})}{p(-1|\boldsymbol{x})} = \frac{U'(F^*)}{U'(-F^*)}$$

を満たす. ここで

$$\rho_U(z) = \frac{1}{2}\log\frac{U'(z)}{U'(-z)}$$

とすると

$$F^*(\boldsymbol{x}) = \rho_U^{-1}\left(\frac{1}{2}\log\frac{p(1|\boldsymbol{x})}{p(-1|\boldsymbol{x})}\right)$$

と表される. 今 $U(z)$ は単調増加関数であるから $\rho_U(z)$ は奇関数で $\rho_U(0) = 0$ を満たすので, $\mathrm{sgn}(F^*(\boldsymbol{x}))$ はベイズルールと等価である.

今 $\rho(z)$ を適当な奇関数として条件付確率が

$$p_\rho(y|\boldsymbol{x};\boldsymbol{\alpha}) = \frac{1}{1+\exp(-2\rho(y\sum_{t=1}^T \alpha_t f_t(\boldsymbol{x})))}$$

を満たすと仮定し, モデルを $M[\rho] = \{p_\rho(y|\boldsymbol{x};\boldsymbol{\alpha})\}$ とする. ロス $U(z)$ に関連したモデルを $M[\rho_L]$ とし, 真の確率構造が $p_{\rho_L}(y|\boldsymbol{x};\boldsymbol{\alpha})$ であるときアブストラクトなロスは $\boldsymbol{\alpha}$ において最小化される. つまりロス $U(z)$ による $\boldsymbol{\alpha}$ の推定量はフィッシャー一致推定量である.

また異なるロス $U_1(z)$,$U_2(z)$ に対して $\rho_{U_1} = \rho_{U_2}$ が成立するならばこれらのロスに関連するモデルは同一である. 例えば

$$U_1(z) = \exp(z), U_2(z) = \log(1+\exp(2z)), U_3(z) = \begin{cases} z & z \geq 0 \\ \frac{1}{2}\exp(2z) - \frac{1}{2} & z < 0 \end{cases},$$

とすると関連するモデルはロジスティックモデルとなる (Eguchi and Copas, 2002).

$$\rho(z) = z, p_0(y|\boldsymbol{x};\boldsymbol{\alpha}) = \frac{1}{1+\exp(-2y\sum_{t=1}^T \alpha_t f_t(\boldsymbol{x}))}.$$

ここで $U_1(z)$ はアダブーストを導くロスであり, $U_2(z)$ はロジットブースト (Friedman et al., 2000), $U_3(z)$ はマダブースト (Domingo and Watanabe, 2000) を導くロスである.

4.2 モスト B-ロバストなロス

今 $F_0(\boldsymbol{x})$ は既知とし 1 パラメータのモデル $M_0[\rho, f] = \{p_\rho^0(y|\boldsymbol{x};\alpha)\}$ を考える. ただし

$$p_\rho^0(y|\boldsymbol{x};\alpha) = \frac{1}{1+\exp(-2\rho(yF_0(\boldsymbol{x}) + \alpha y f(\boldsymbol{x})))}$$

とする．$\rho(z)=z$ とするとロジスティックモデルとなり，上記のモデルはモデルがロジスティックモデルとどのくらい離れているかを $\rho(z)$ によって記述している．外れ値に対するロバストさを測る指標としてグロスエラーセンシティビティを考える．真の分布を $p^0_\rho(y|\boldsymbol{x};\alpha_0)p(\boldsymbol{x})$ として $(\tilde{\boldsymbol{x}},\tilde{y})$ に外れ値がある時の汚染分布を

$$p'(\boldsymbol{x},y)=(1-\epsilon)p(\boldsymbol{x})p^0_\rho(y|\boldsymbol{x};\alpha_0)+\epsilon\delta(\tilde{\boldsymbol{x}},\tilde{y})$$

とする．ただし ϵ は汚染が起こる確率とする．この時汚染分布の下でアブストラクトなロスを最小にする推定量を

$$\alpha_\epsilon(\tilde{\boldsymbol{x}},\tilde{y})=\operatorname*{argmin}_\alpha \mathrm{E}_{p'(\boldsymbol{x},y)}[U(-yF_0(\boldsymbol{x})-\alpha yf(\boldsymbol{x}))]$$

とする．グロスエラーセンシティビティは $\alpha_\epsilon(\tilde{\boldsymbol{x}},\tilde{y})$ を用いて以下の様に定義される．

$$\gamma(L,\alpha_0)=\sup_{\tilde{\boldsymbol{x}},\tilde{y}}\lim_{\epsilon\to+0}\left(\frac{\alpha_\epsilon(\tilde{\boldsymbol{x}},\tilde{y})-\alpha_0}{\epsilon}\right)^2$$

つまり分布が汚染されたときに，ロス $U(z)$ を用いた推定量がどの程度ずれやすいかを最悪評価した量となっている．

定理 2 :$M_0[\rho.h]$ に対するモスト B-ロバストなロスは以下の様に表される．

$$U_\rho(z)=\begin{cases} z & z\geq 0 \\ \int_0^z \exp(2\rho(w))\mathrm{d}w & z<0 \end{cases} \tag{23}$$

またこのロスから導かれるアルゴリズムの重みは以下のように表される．

$$\begin{aligned} U'_\rho(z) &= \begin{cases} 1 & z\geq 0 \\ \exp(2\rho(z)) & z<0 \end{cases} \\ w_t(i) &\propto U'_\rho(-F_{t-1}(\boldsymbol{x}_i)y_i) \end{aligned} \tag{24}$$

証明: 汚染された分布の下での推定量 α' は以下を満たす．

$$\begin{aligned} 0 &= (1-\epsilon)\mathrm{E}_{p(\boldsymbol{x})p^0_\rho(y|\boldsymbol{x};\alpha_0)}[U'(-yF_0(\boldsymbol{x})-\alpha' yf(\boldsymbol{x}))yf(\boldsymbol{x})] \\ &\quad +\epsilon U'(-\tilde{y}F_0(\tilde{\boldsymbol{x}})-\alpha'\tilde{y}f(\tilde{\boldsymbol{x}}))\tilde{y}f(\tilde{\boldsymbol{x}}). \end{aligned} \tag{25}$$

Hampel et al.(1986) の議論に従い計算すると近似的に

$$\alpha'(\tilde{\boldsymbol{x}},\tilde{y})-\alpha_0=\frac{\epsilon U'(-\tilde{y}F_0(\tilde{\boldsymbol{x}})-\alpha_0\tilde{y}f(\tilde{\boldsymbol{x}}))\tilde{y}f(\tilde{\boldsymbol{x}})}{(1-\epsilon)\mathrm{E}_{p(\boldsymbol{x})p^0_\rho(y|\boldsymbol{x};\alpha_0)}[U''(-yF_0(\boldsymbol{x})-\alpha_0 yf(\boldsymbol{x}))]+\epsilon U''(-\tilde{y}F_0(\tilde{\boldsymbol{x}})-\alpha_0\tilde{y}f(\tilde{\boldsymbol{x}}))}$$

が得られ，グロスエラーセンシティビティは

$$\gamma(L,\alpha_0)=\left[\frac{\sup_{(\tilde{\boldsymbol{x}},\tilde{y})}U'(-\tilde{y}F_0(\tilde{\boldsymbol{x}})-\alpha_0\tilde{y}f(\tilde{\boldsymbol{x}}))}{\mathrm{E}_{p(\boldsymbol{x})p^0_\rho(y|\boldsymbol{x};\alpha_0)}[U''(-yF_0(\boldsymbol{x})-\alpha_0 yf(\boldsymbol{x}))]}\right]^2$$

となる．また $\rho_U(z)$ の微分から以下の関係式が得られる．

$$U''(z) = 2\rho'(z)U'(-z)e^{2\rho(z)} - U''(-z)e2\rho(z).$$

この関係式を用いると

$$\gamma(L,\alpha_0) = \left[\frac{\sup_{(\tilde{\boldsymbol{x}},\tilde{y})} U'(-\tilde{y}F_0(\tilde{\boldsymbol{x}}) - \alpha_0\tilde{y}f(\tilde{\boldsymbol{x}}))}{2\int_{\boldsymbol{x}} p(\boldsymbol{x})\rho'(F_0(\boldsymbol{x}) + \alpha_0 f(\boldsymbol{x}))U'(-F_0(\boldsymbol{x}) - \alpha_0 f(\boldsymbol{x}))p_\rho^0(1|\boldsymbol{x};\alpha_0)}\mathrm{d}\boldsymbol{x}\right]^2 \quad (26)$$

以下が得られる．ところで

$$\sup_{(\tilde{\boldsymbol{x}},\tilde{y})} U'(-\tilde{y}F_0(\tilde{\boldsymbol{x}}) - \alpha_0\tilde{y}f(\tilde{\boldsymbol{x}})) = \infty$$

が成立する場合はグロスエラーセンシティビティも発散してしまう．そこでロスを定数倍しても導かれるアルゴリズムや関連するモデル等は変わらない事に注意して

$$\sup_{(\tilde{\boldsymbol{x}},\tilde{y})} U'(-\tilde{y}F_0(\tilde{\boldsymbol{x}}) - \alpha_0\tilde{y}f(\tilde{\boldsymbol{x}})) = 1$$

となるロスについてのみ考察する．したがって (26) の分母を最大にすればよいことになり，分母は $L'(z) = L'_\rho(z)$ によって最大化される．したがってグロスエラーセンシティビティを最小にするロスは (23) で与えられる．

具体的な例についてモスト B-ロバストなロスを考えてみる．ロジスティックモデル，つまり $\rho(z) = z$ の場合，

$$p(y|\boldsymbol{x}) = \frac{1}{1+\exp(-2yF(\boldsymbol{x}))}$$

に対するモスト B-ロバストなロスを考える．このモデルに関連するロスとしてはアダブーストやロジットブーストがあるが，モスト B-ロバストなロスはマダブースト (Domingo and Watanabe, 2000) を導くロス

$$L_\rho(z) = L_{Mada}(z) = \begin{cases} z & z \geq 0 \\ \frac{1}{2}(\exp(2z) - 1) & z < 0 \end{cases}$$

である (図 4)．図 5 はアルゴリズムの重みに対応するロスの導関数を示している．導関数のグラフを見ると $z > 0$ の部分，つまり間違えた例題に対する重みに対応する部分が定数となっており，判別しにくい外れ値に対して重みをかけすぎない様になっている．

5 モスト B-ロバストな η-ブースト

前章までで，y の外れ値，つまりミスラベルに対してロバストなブーストアルゴリズムとして汚染モデルを考慮した η-ブースト，また $\boldsymbol{x}$ における外れ値を係数の推定の立場から影響関数を尺度としてロバスト化したモスト B-ロ

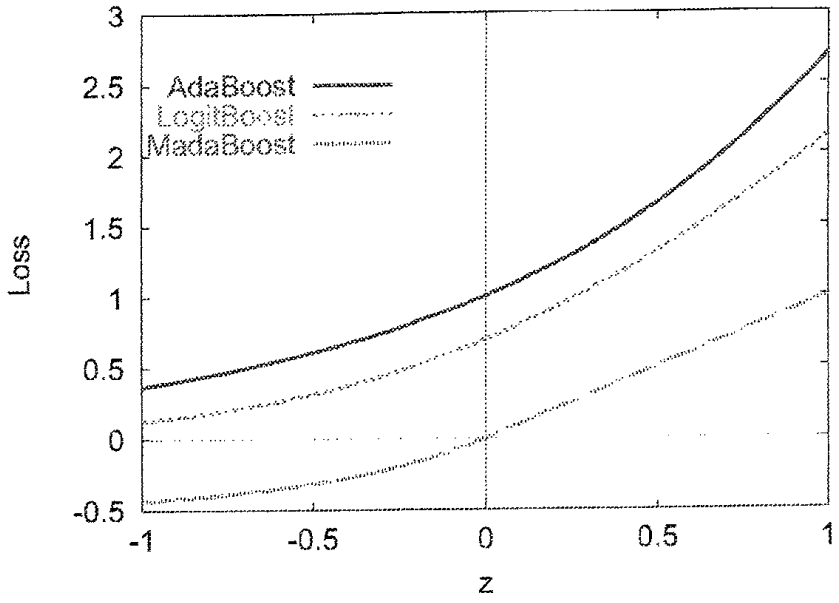

図 4: ロジスティックモデルに関連したロスのグラフ.

バストなロスを提案した. 本章ではミスラベルにも $\boldsymbol{x}$ における外れ値に対してもロバストなロスを考案する.

η-ブーストは $U_\eta(z) = (1-\eta)\exp(z) + \eta z$ の逐次最小化から得られ, 対応するモデルは (12) であり, 汚染モデル (21) として解釈する事ができた. このモデルは境界付近でミスラベルの確率が高くなり, 境界から離れれば離れるほどミスラベルの確率は低くなる. この汚染モデルに対応するロスでモスト B-ロバストなロスは, 前章の考察から以下の形となる.

$$U_{\eta,\rho}(z) = \begin{cases} z, & z \geq 0, \\ \frac{(1-\eta)(\exp(z)-1)\eta + (2\eta-1)\log(1+(\exp(z)-1)\eta)}{\eta^2}, & z < 0. \end{cases}$$

またアルゴリズムの重みに対応するロスの導関数は

$$U'_{\eta,\rho}(z) = \begin{cases} 1 & z \geq 0, \\ \frac{(1-\eta)\exp(z)+\eta}{(1-\eta)\exp(-z)+\eta} & z < 0, \end{cases}$$

となる. このロスは η を 0 に近づけるとマダブーストのロスとなる.

$$\lim_{\eta \to 0} U_{\eta,\rho}(z) = U_{mada}.$$

このロスから導かれるブースティングアルゴリズムは y の外れ値 (ミスラベル) にも $\boldsymbol{x}$ の外れ値 (グロスエラーセンシティビティ) の意味でもロバストとなる. 図 6, 図 7 はそれぞれロスとその導関数を示している.

ミスラベルの度合いを表す η としてどの値を用いるべきかは通常の η-ブーストの場合と同様にクロスバリデーションなどを用いて決めることが出来る.

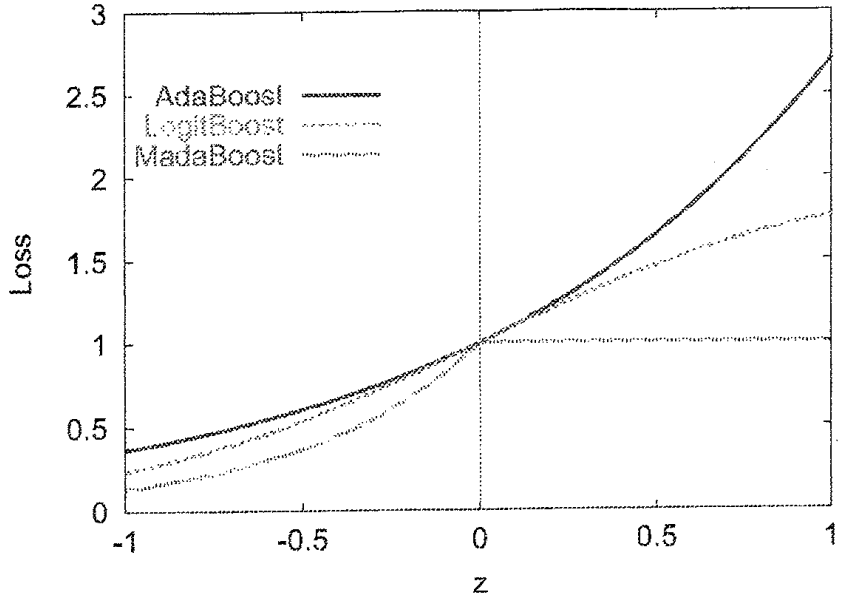

図 5: $U'(z)$（アルゴリズムの重み）のグラフ.

6 結論

本稿ではアダブーストの外れ値に弱いという欠点を克服するために，$\boldsymbol{x}$，y における外れ値の性質の違いに着目しロバストなロスに関して考察した．つまり

1. y における外れ値に対してロバストにするためにミスラベルを考慮した汚染モデルに関連したロスから η-ブーストを導き，
2. 係数の推定の意味でグロスエラーセンシティビティを指標としてモスト B-ロバストなロスを導出し，
3. 上記の二つを組み合わせて $\boldsymbol{x}$,y どちらの外れ値に対してもロバストなロスを導いた．

参考文献

Amari, S. and Nagaoka, H. (2000). *Methods of Information Geometry*. Oxford University Press

Copas, J. (1988). Binary Regression Models for Contaminated Data. *J. Royal Statist. Soc. B*, Vol. 50, 225-265.

Domingo, C. and Watanabe, O. (2000). MadaBoost: A modification of AdaBoost. In *Proc. of the 13th Conference on Computational Learning Theory*.

Eguchi, S. and Copas, J. (2002). A class of logistic type discriminant functions. *Biometrika*, **89**, 1-22.

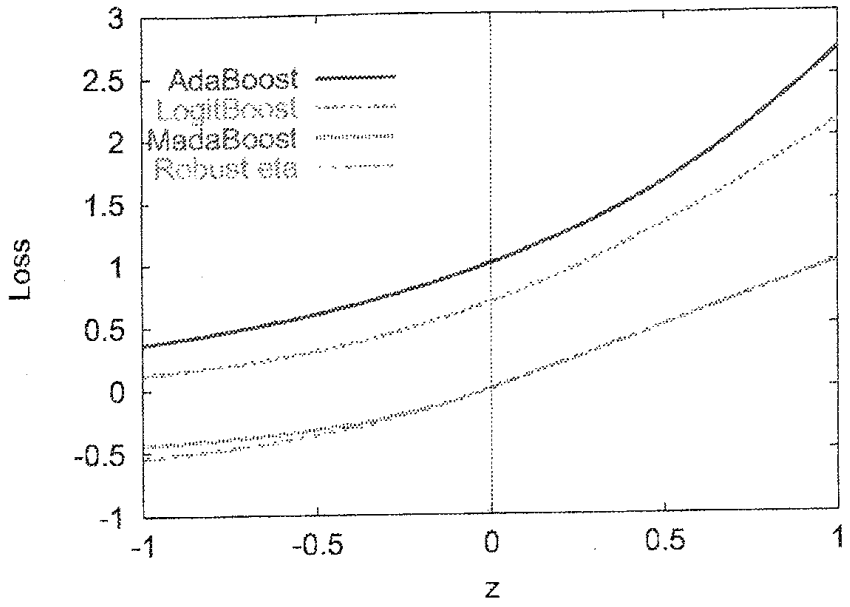

図 6: モストB-ロバストな η-ブーストのロスのグラフ.

Eguchi, S. and Copas, J. (2001). Recent developments in discriminant analysis from an information geometric point of view. *J. Korean Statist. Soc.*, **30**, 247-264.

Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *J. Computer and System Sciences*, **55**, 119-139.

Friedman, J., Hastie, T. and Tibishirani, R. (2000). Additive logistic regression: A statistical view of boosting. *Ann. Statist.*, **28**, 337-407.

Hampel, F. R., Rousseeuw, P. J., Ronchetti, E. M. and Stahel, W. A. (1986). *Robust Statistics: the Approach Based on Influence Functions.* Wiley, New York.

Hastie, T. Tibishirani, R. and Friedman, J. (2001). *The elements of statistical learning.* Springer, New York.

Kanamori T., Takenouchi T., Eguchi S. and Murata N. (2004) The most robust loss function for boosting. In *Neural Information Processing: 11th International Conference, ICONIP*, Lecture Notes in Computer Science , 496-501.

Lebanon, G. and Lafferty, J. (2001). Boosting and maximum likelihood for exponential models. *Advances in Neural Information Processing Systems* , **14**.

Mason, L. Baxter, J. Bartlett, P. and Frean, M. (1999). Boosting Algorithms as Gradient Decent in Function Space. *Advance in Neural Information Processing Systems*, **11**.

Mclachlan, G. (1992). *Discriminant analysis and statistical pattern recognition.* Wiley, New York.

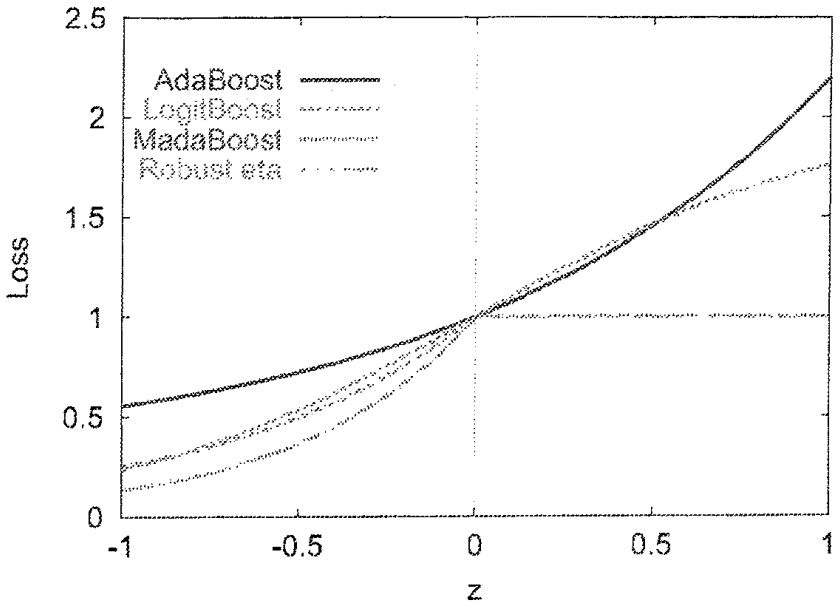

図 7: ロスの導関数のグラフ.

Murata, N.,Takenouchi, T., Kanamori, T. and Eguchi, S. (2004). Information geometry of U-boost and Bregman divergence. *Neural Computation*, **16**, 1437-1481.

Rätsch, G., Onoda, T. and Müller K.-R. (2001) Soft Margins for AdaBoost. *Machine Learning*, **42**, 287-320.

Schapire, R. (1990). The strength of weak learnability. *Machine Learning*, **5** , 197-227.

Shapire, R. (1999). Theoretical View of Boosting. In *Proc. of the 4th European Conference on Computational Learning Theory.*

Takenouchi, T. and Eguchi, S. (2004). Robustifying AdaBoost by adding the naive error rate. *Neural Computation*, **16**, 767-787.