カステラ本10章

10.1

ブースティング法

- 考え方の始まり
  - 大量の弱分類器の出力を組み合わせて強力な出力を得る
- もともとは分類問題のために設計されていたが、回帰問題にも拡張できる

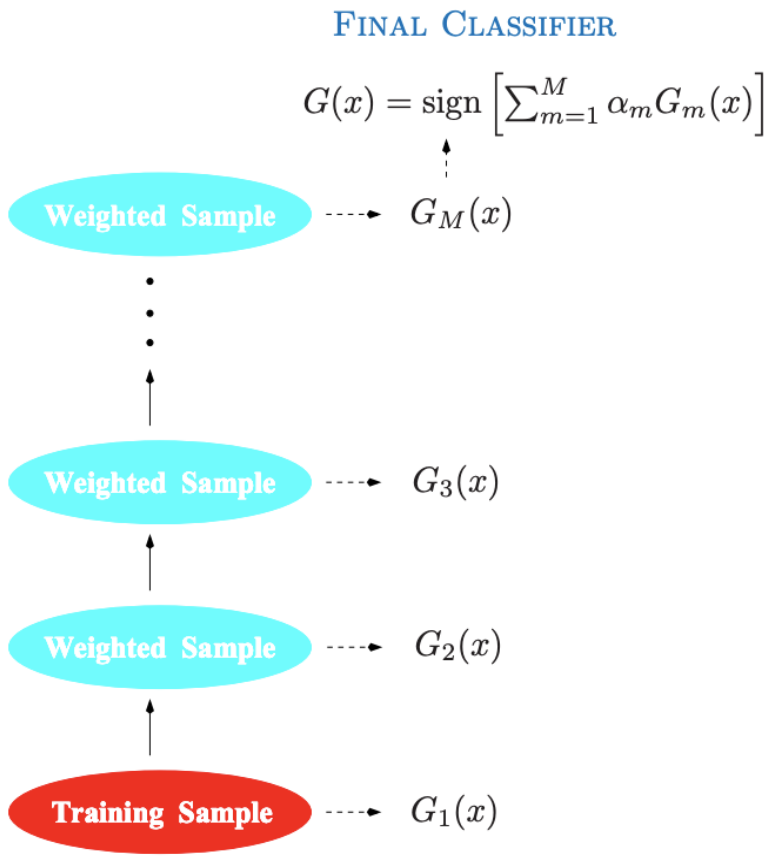
アダブーストM1（2クラス分類）

$$G(x) = sign(\sum_{m=1}^{M} \alpha_m G_m(x))$$

$sign()$：$\{-1, 1\}$の予測結果を出力

α_m：$G_m(x)$ の貢献度

※より正確な分類器に高い貢献度を付与



アダブーストM1のアルゴリズム(10.1)

1. 観測値に付加する重みを初期化：$w_i = \frac{1}{N}(i = 1, 2, ..., N)$
2. $m = 1, ...M$に対して以下を行う
   (a) 重み w_i を用いて分類器 $G_m(x)$ を学習データに当てはめる
   (b) (a)の結果から得られる重み付き誤分類率を計算：$err_m = \frac{\sum_{i=1}^{N} w_i I(y_i \neq G_m(x_i))}{\sum_{i=1}^{N} w_i}$
   (c) $G_m(x)$に対する重み（影響力）の計算： $\alpha_m = log(\frac{1-err_m}{err_m})$
   (d) 観測値の重み w_iを以下の通り更新

$$w_i \leftarrow w_i * exp[\alpha_m \times I(y_i \neq G_m(x))](i = 1, 2, ..., N)$$

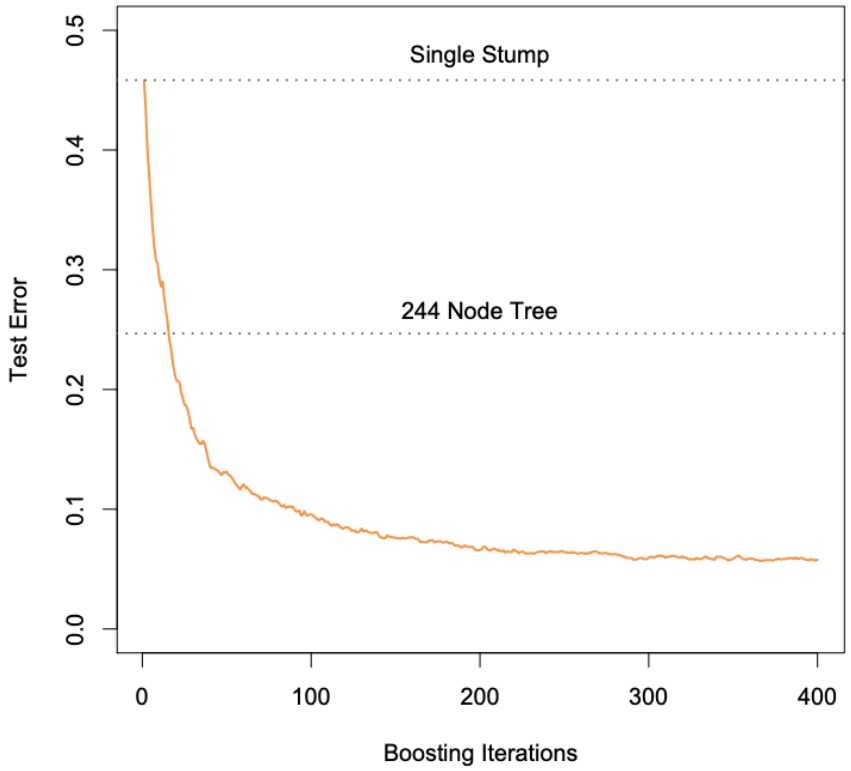
3. 予測値 $G(x) = sign[\sum_{m=1}^{M} \alpha_m G_m(x)]$ を出力

アダブーストによる誤分類率の精度向上の様子

- 特徴量 $X_1, \ldots, X_{10}$

$$Y = \begin{cases} 1 & \sum_{j=1}^{10} X_j^2 > \chi_{10}^2(0.5) = 9.34 \\ -1 & \text{その他} \end{cases}$$

- **非常に単純な弱分類器をブースティングすることで、予測誤差は単独の大きな分類木に比べて大幅に改善されている**



10章の展開(前半部分)

- 10.2〜10.4
  - アダブーストが基本学習器の加法的モデルを当てはめており、（負の）二項対数尤度と似た指数損失関数を最適化している
- 10.5
  - 指数損失関数の母集団での最小化が、クラス帰属確立の対数オッズとなる
- 10.6
  - 二乗誤差や指数損失よりロバストな、回帰や分類のための損失関数について
- 10.7, 10.9
  - ブースティングのデータマイニング応用において、決定木が理想的な学習器であることについて
- 10.8
  - 手法の応用例

10.2

ブースティングによる加法的モデル当てはめ

ブースティングの何がすごいのか

- 基底関数 $G_m(x)$ を用いた加法的展開の当てはめでありながら、その計算コストの高さを改善する形で適用できる

$$\min_{\{\beta_m,\gamma_m\}_1^M} \sum_{i=1}^{N} L(y_i, \sum_{m=1}^{M} \beta_m b(x_i;\gamma_m))$$

- $\min_{\beta,\gamma} \sum_{i=1}^{N} L(y_i, \beta b(x_i;\gamma))$ を高速に解くことができれば、代替手法により簡略化できる

10.3

前向き段階的加法的モデリング

- 既に追加された基底関数のパラメータや係数を調整することなく、
  新たな基底関数を展開に順次追加していく
- m 回目の繰り返しにおいて、現在の展開 $f_{m-1}(x)$ に追加するための、
  最適な基底関数 $b(x;\gamma_m)$ と係数 β_m を求めることで $f_m(x)$ を生成する
- 以前に追加された項を修正しない形で上記プロセスを繰り返す

前向き段階的加法的モデリングのアルゴリズム(10.2)

1. 初期化：$f_0(x) = 0$
2. $m = 1, ..., M$ に対して以下を行う

(a) 次の計算をする

$$(\beta_m, \gamma_m) = \arg\min_{\beta,\gamma} \sum_{i=1}^{N} L(y_i, f_{m-1}(x_i) + \beta b(x_i; \gamma))$$

(b) 展開を更新する

$$f_m(x) = f_{m-1}(x) + \beta_m b(x; \gamma_m)$$

10.4

指数損失とアダブースト

- **アダブーストM1は、損失関数 $L(y, f(x)) = exp(-yf(x))$ (10.8)を用いた前向き段階的加法的モデリングである**
- アルゴリズム(10.2) 2(a)より、

$$(\beta_m, G_m) = \arg\min_{\beta, G} \sum_{i=1}^{N} w_i^{(m)} exp(-\beta y_i G(x_i)) \qquad (10.9)$$

- ここで、$w_i^{(m)} = exp(-y_i f_{m-1}(x_i))$
  - β にも $G(x)$ にも依存しない各観測値に対する重み
  - f_{m-1} に依存するため、それぞれの重みは繰り返し回数 m とともに変化する

学習データへの当てはめ

- y の予測における重み付き誤差率を最小化していると見ることができ、
  以下のように表現できる

$$e^{-\beta} \sum_{y_i = G(x_i)} w_i^{(m)} + e^{\beta} \sum_{y_i \neq G(x_i)} w_i^{(m)}$$

$$\rightarrow (e^{\beta} - e^{-\beta}) \sum_{i=1}^{N} w_i^{(m)} I(y_i \neq G(x_i)) + e^{-\beta} \sum_{i=1}^{N} w_i^{(m)} \qquad (10.11)$$

- つまり、アルゴリズム(10.1)における

> (a) 重み w_i を用いて分類器 $G_m(x)$ を学習データに当てはめる

は、式(10.10)と(10.11)における最小化の近似と見ることができる

(10.9) 式を解くために

- 任意の β に対する G_m
  - 予測値 y の重み付き誤差率を最小化する分類器

$$G_m = \arg\min_G \sum_{i=1}^{N} w_i^{(m)} I(y_i \neq G(x_i)) \qquad (10.10)$$

- β_m の推定値 (G_mを式(10.9)に代入し、βについて解く)

$$\beta_m = \frac{1}{2} log \frac{1 - err_m}{err_m}$$

- 最小化重み付き誤差率：$err_m = \frac{\sum_{i=1}^{N} w_i^{(m)} I(y_i \neq G_m(x_i))}{\sum_{i=1}^{N} w_i^{(m)}}$

3. モデルの更新

- 近似式

$$f_m(x) = f_{m-1}(x) + \beta_m G_m(x)$$

- 重みの更新

$$w_i^{(m+1)} = w_i^{(m)} e^{-\beta_m y_i G_m(x_i)} \qquad (10.14)$$

3. モデルの更新

- $-y_i G_m(x_i) = 2 \cdot I(y_i \neq G_m(x_i))$ により式(10.14)を変換すると

$$w_i^{(m+1)} = w_i^{(m)} e^{2\beta_m I(y_i \neq G_m(x_i))} e^{-\beta_m}$$

- 分類器（G_m）に対する重み（影響力）の計算（アルゴリズム10.1 2(c)）

$$2\beta_m = log \frac{1 - err_m}{err_m} = \alpha_m$$

- 重みを更新

$$w_i^{(m+1)} = w_i^{(m)} e^{\alpha_m I(y_i \neq G_m(x_i))} e^{-\beta_m} \qquad (10.15)$$

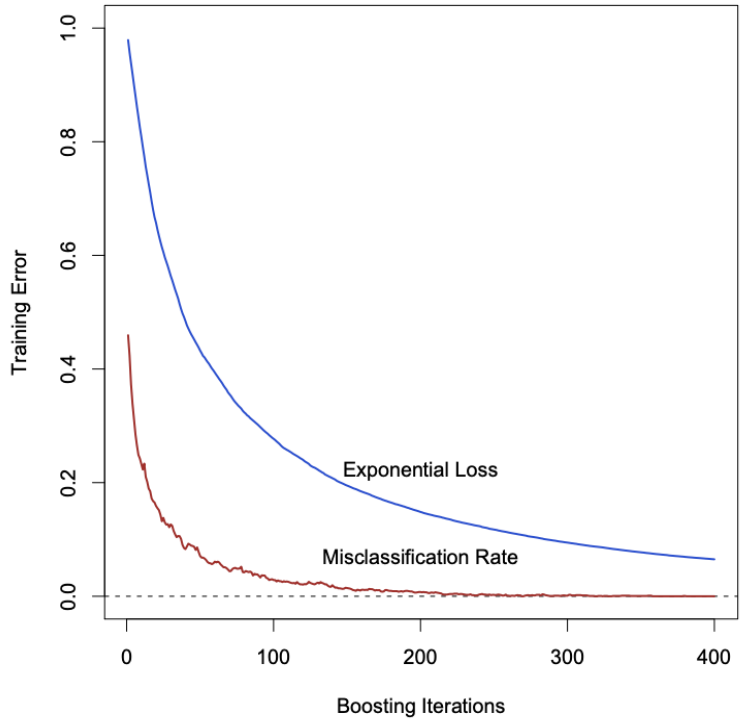
3. モデルの更新

- 式(10.15)は、アルゴリズム(10.1) 2(d)

> $$w_i \leftarrow w_i * exp[\alpha_m \times I(y_i \neq G_m(x))](i = 1, 2, ..., N)$$

と等価といえる
（$e^{-\beta_m}$ は、全ての重みに
同じ値をかけるため、効果なし）

- アダブーストは訓練集合に対する誤分類率を
最適化しているわけではない



10.5

なぜ指数損失関数か

- 加法的モデリングの観点での指数損失の魅力
  - 計算効率性
  - 統計的性質における重要性

指数損失の統計的重要性

- 指数損失が何を推定するのか
  - 母集団での最小化を考える

$$f^{\star}(x) = \underset{f(x)}{\arg\min}\, E_{Y|x}(e^{-Yf(x)}) = \frac{1}{2} log \frac{P_r(Y=1|x)}{P_r(Y=-1|x)} \qquad (10.16)$$

- アダブーストにより得られる加法的展開は、$P(Y=1|x)$ の対数オッズの2分の1を推定している
- さらに、(10.16)と等価な以下が成り立つ

$$P_r(Y=1|x) = \frac{1}{1+e^{-2f^{\star}(x)}}$$

指数損失の統計的重要性

- f をロジット変換と捉えると、母集団での最小化に対する別の損失規準として、**逸脱度**を考えることができる
- 逸脱度の最小化を規定する真の確率$p(x)$

$$p(x) = P_r(Y = 1|x) = \frac{e^{f(x)}}{e^{-f(x)} + e^{f(x)}} = \frac{1}{1 + e^{-2f(x)}} \qquad (10.17)$$

- よって、母集団での$E_{Y|x}\left[e^{-Yf(x)}\right]$ の最小化と逸脱度が同じになるため、2つの規準が同じ解にたどり着く

10.6 損失関数とロバスト性

分類のためのロバスト損失関数

- 指数損失：

$$L(y, f(x)) = exp(-yf(x)) \qquad (10.8)$$

- 逸脱度：

$$-l(y, f(x)) = log(1 + e^{-2Yf(x)}) \qquad (10.18)$$

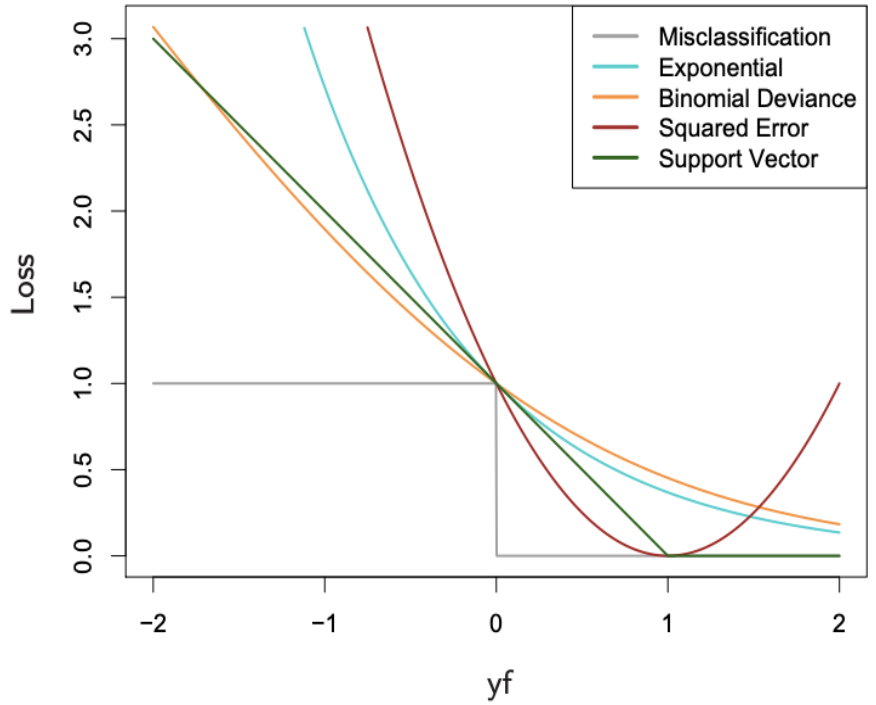
- マージン = $-yf(x)$
- 分類問題において、マージンは回帰における残差$y - f(x)$と同じ役割を持つ

分類のためのロバスト損失関数

- 分類問題におけるマージンの振る舞い
  - 分類器 $G(x) = sign[f(x)] \qquad \in \{-1, 1\}$
    - $y_i f(x_i) > 0$ で正しく分類（$y_i > 0, f(x) > 0 / y_i < 0, f(x) < 0$）
    - $y_i f(x_i) < 0$ で誤分類（$y_i > 0, f(x) < 0 / y_i < 0, f(x) > 0$）
    - 決定境界：$f(x) = 0$
  - 分類アルゴリズムの目的
    - 負のマージンに対して罰則を与える
    - **正のマージンは既に正しく分類できているので、罰則はいらない**

各損失関数の特徴比較

- 指数 / 逸脱度：
  - 罰則の与え方の度合い
    - 指数は大きな負のマージンに強い影響を受ける
    - 逸脱度の方が全てのデータに均等に影響を広げられるため、よりロバストになる



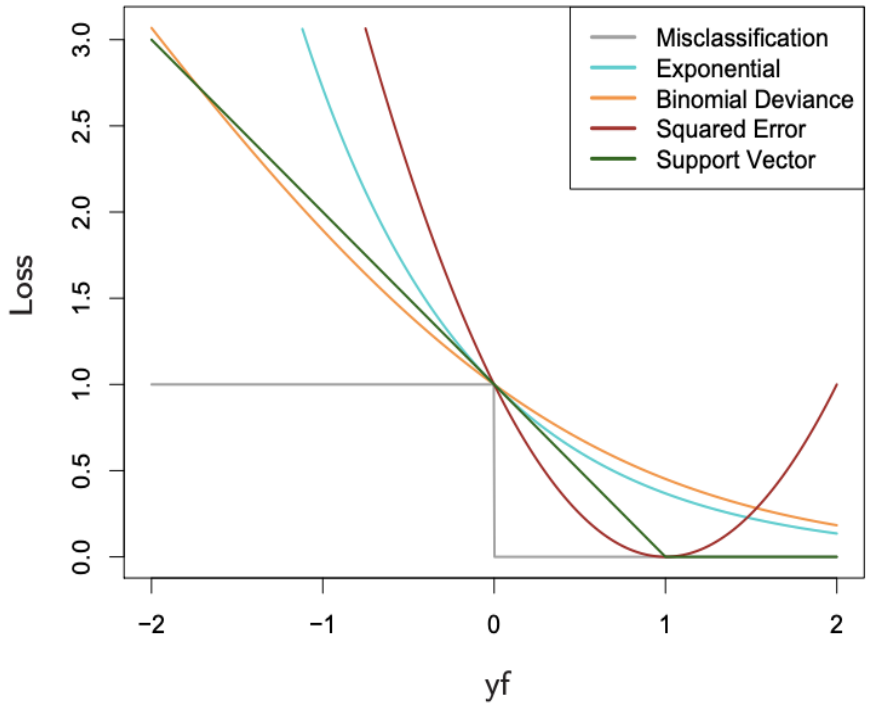
各損失関数の特徴比較

- 二乗誤差：
  - 一部領域において精度が高い

$$f^{\star}(x) = \underset{f(x)}{\arg\min}\, E_{Y|x}(Y - f(x))^2$$
$$= E(Y|x) = 2P_r(Y = 1|x) - 1$$

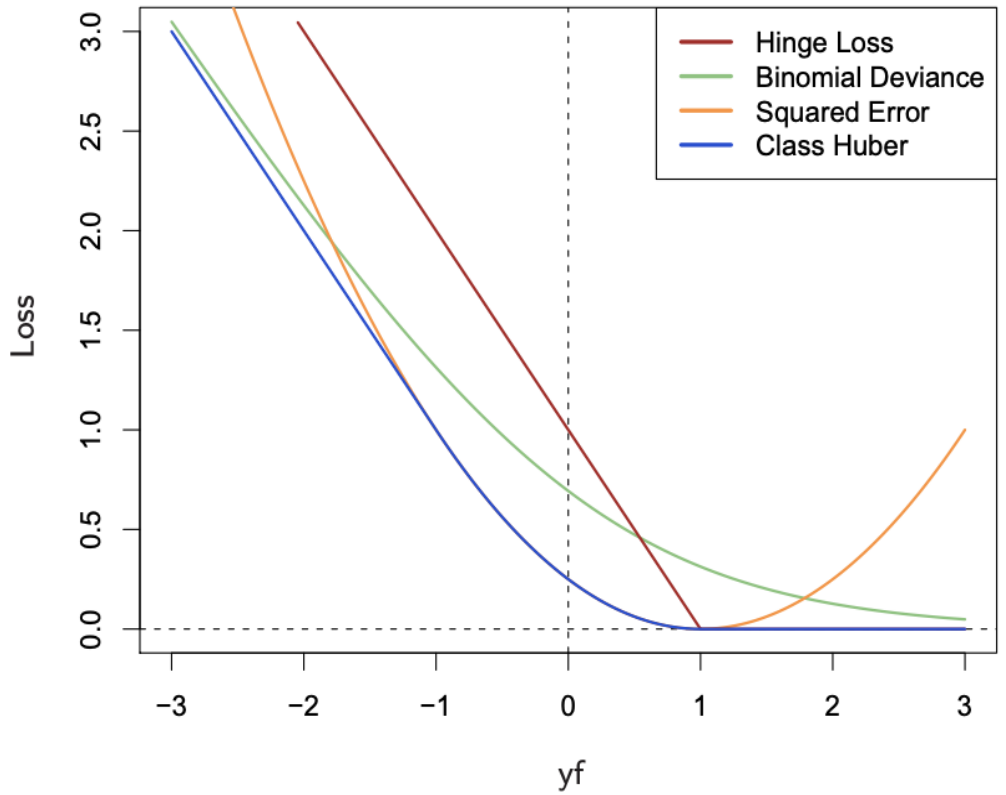
- ヒンジ損失
  - 正のマージンに対する罰則を0にできる

$$L(y, f(x)) = \begin{cases} 0 & \text{if} \quad yf(x) > 1 \\ \text{線形} & \text{if} \quad yf(x) < -1 \end{cases}$$



各損失関数の特徴比較

- フーバー的2乗ヒンジ損失 (図12.4)
  - 逸脱度、2乗誤差、ヒンジ損失の良い部分を組み合わせる



多クラス分類

- 応答Y：　　$\mathcal{G} = \{\mathcal{G}_1, ..., \mathcal{G}_k\}$
- クラス所属確率：　　$p_k(x) = \Pr(Y = \mathcal{G}|x), \quad k = 1, ..., K$
- クラス所属確率のロジスティックモデル：　　$p_k(x) = \frac{e^{f_k(x)}}{\sum_{l=1}^{K} e^{f_l(x)}}$
- $\quad 0 \leq p_k(x) \leq 1 \quad$ で $\quad \sum_{k=1}^{K} p_k(x) = 1$
- $f_k(x)$ には冗長性があり、対称性を保つことが望ましいため、
  次の拘束条件を課す：　　$\sum_{k=1}^{K} f_k(x) = 0$

多クラス分類

- クラス推定：ベイズ分類器

$$G(x) = \mathcal{G}_k \quad where \quad k = \arg\max_k p_l(x)$$

- 損失関数も多クラスに自然に拡張できる
  - 逸脱度：

$$L(y, p(x)) = -\sum_{k=1}^{K} I(y = \mathcal{G}_k) log p_k(x)$$

$$= -\sum_{k=1}^{K} I(y = \mathcal{G}_k) f_k(x) + log(\sum_{l=1}^{K} e^{f_l(x)})$$

回帰のためのロバスト損失関数

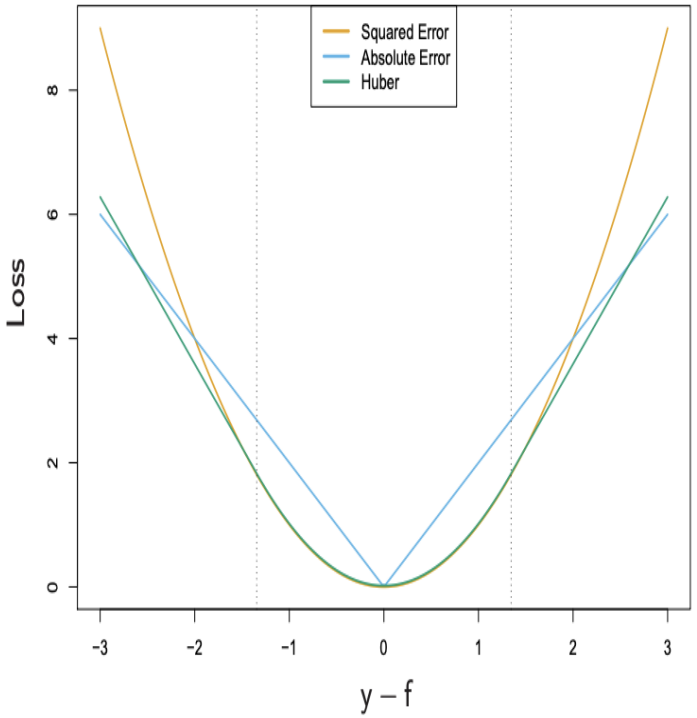
損失関数	式	母集団乗の解	有限サンプル上
2乗誤差損失	$L(y, f(x)) = (y - f(X))^2$	$f(x) = E(Y\|x)$	大きな絶対残差$\|y_i - f(x_i)\|$を伴う観測値に対して、極端に大きな重みを与える→ ロバスト性の低下
絶対値損失	$L(y, f(x)) = \|y - f(X)\|$	$f(x) = median(Y\|x)$	ガウス誤差に対する最小二乗損失とほぼ同等の有効性を持ちつつ、ロバスト性を持つ

回帰のためのロバスト損失関数

絶対値誤差損失の特徴を拡張

- フーバー損失

$$L(y, f(x)) = \begin{cases} [y - f(x)]^2 & for \quad |y - f(x)| \leq \delta \\ 2\delta|y - f(x)| - \delta^2 & otherwise \end{cases}$$



ロバスト性まとめ

- ロバスト性に関しては、
  - 回帰における二乗誤差損失
  - 分類における指数損失

    のように、統計学的に最良とは言えない規準もある
- しかし、これらは段階的加法的モデリングにおける

  単純にモデルを当てはめられるという簡潔なアルゴリズムを提示できる

  （これをよりロバストな規準に単純に置き換えることはできない）

10.7

データマイニングの「万能」手法

- ここまでに紹介してきた予測学習法は、特定の状況のもとで最良の性能を出す
- しかし、与えられた問題に対してどれが最良かを事前に予測できることはほとんどない

TABLE 10.1. *Some characteristics of different learning methods. Key:* ▲ *= good,* ◆ *=fair, and* ▼ *=poor.*

Characteristic	Neural Nets	SVM	Trees	MARS	k-NN, Kernels
Natural handling of data of "mixed" type	▼	▼	▲	▲	▼
Handling of missing values	▼	▼	▲	▲	▲
Robustness to outliers in input space	▼	▼	▲	▼	▲
Insensitive to monotone transformations of inputs	▼	▼	▲	▼	▼
Computational scalability (large N)	▼	▼	▲	▲	▼
Ability to deal with irrelevant inputs	▼	▼	▲	▲	▼
Ability to extract linear combinations of features	▲	▲	▼	▼	◆
Interpretability	▼	▼	◆	▲	▼
Predictive power	▲	▲	▼	◆	▲

ビジネス領域でのデータマイニングあるある

- 与えられるデータ量や変数量が非常に多くなる
- データが乱雑
  - データ型が混合
  - 欠損値の存在
  - 予測変数と応答変数の分布が歪んでいることが多い
  - 外れ値を含むことが多い
  - 予測変数間で大きくスケールが異なる

ビジネス領域でのデータマイニングあるある

- 予測変数のうち、一部しか予測の役に立っていないことがある
- 特徴量を選択するための、対象分野についての明確な知識が得られない
- データマイニングでは、一般的に説明性の高いモデルが求められる
    - 説明変数と予測値の関係に関する定性的理解を促す情報がある方が良い

「万能手法」決定木

- 比較的高速に説明性の高いモデルを構築するのに有効
  - 変数型の混合、欠損値の利用に適する
  - 各説明変数の変換に対して不変
  - 外れ値の影響を受けない
  - 木を構成する各段階で、内部的な特徴量選択を行っている
- 決定木をブースティングすることで、決定木の致命的欠点である**予測精度**を大幅に改善できる
- さらに、勾配ブースティングモデルを採用することで、ブースティングにおけるスピードや説明性、アダブーストで問題となっていたロバスト性などに関する
  欠点を緩和できる

10.8

例：スパムデータ

- 9.1.2項で用いたのと同じテストデータに各手法を適用し、誤差率を比較

手法	テスト誤差率
勾配ブースティング	**4.5%**
加法的ロジスティック回帰	5.5%
CART	8.7%
MARS	5.5%

例：スパムデータ

- 予測変数の相対的重要度
- ここでモデリングされている量は

$$f(x) = log\frac{\Pr(spam|x)}{\Pr(email|x)}$$

- 最もスパムと関連の強い予測変数
  - `!`, `remove`, `edu`, `hp`

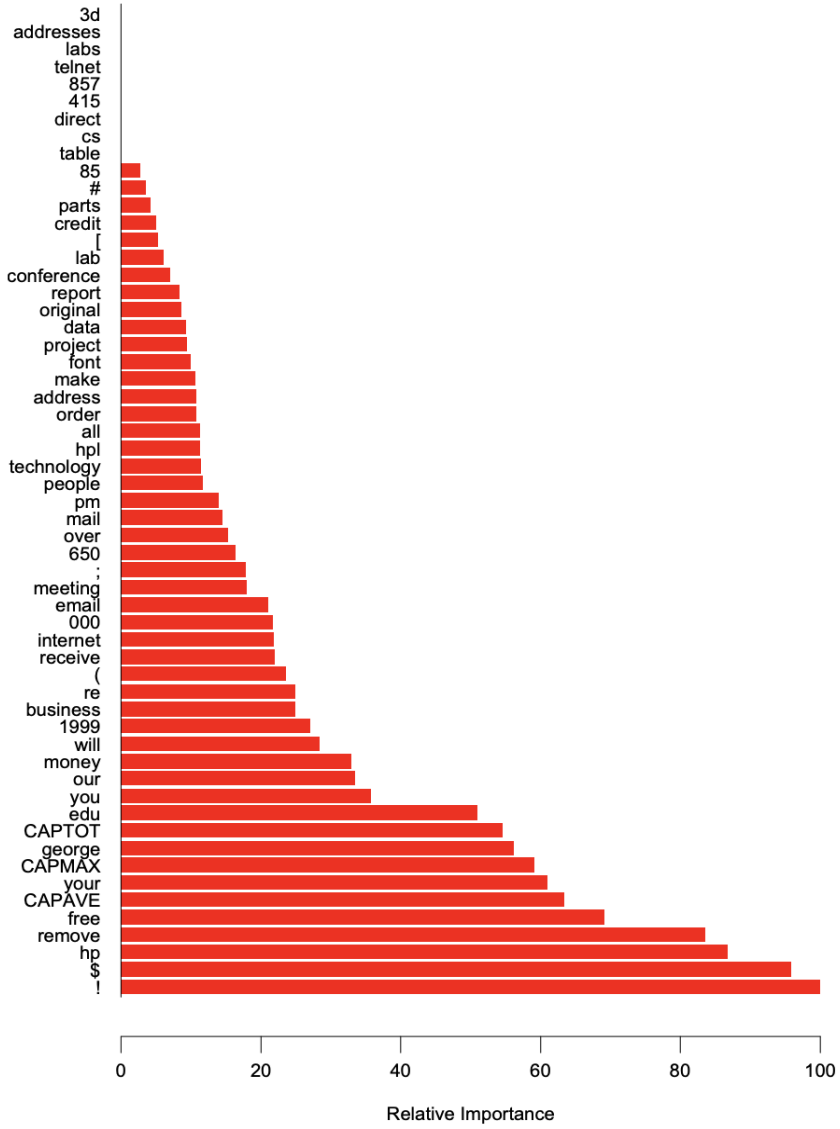


FIGURE 10.6. *Predictor variable importance spectrum for the* `spam` *data. The variable names are written on the vertical axis.*

例：スパムデータ

- スパムとの関連性
  - 正の関連：`!` / `remove`
  - 負の関連：`edu` / `hp`
- それぞれの依存性関数は、加法的ロジスティック回帰モデルにより得られる関数と同様の傾向

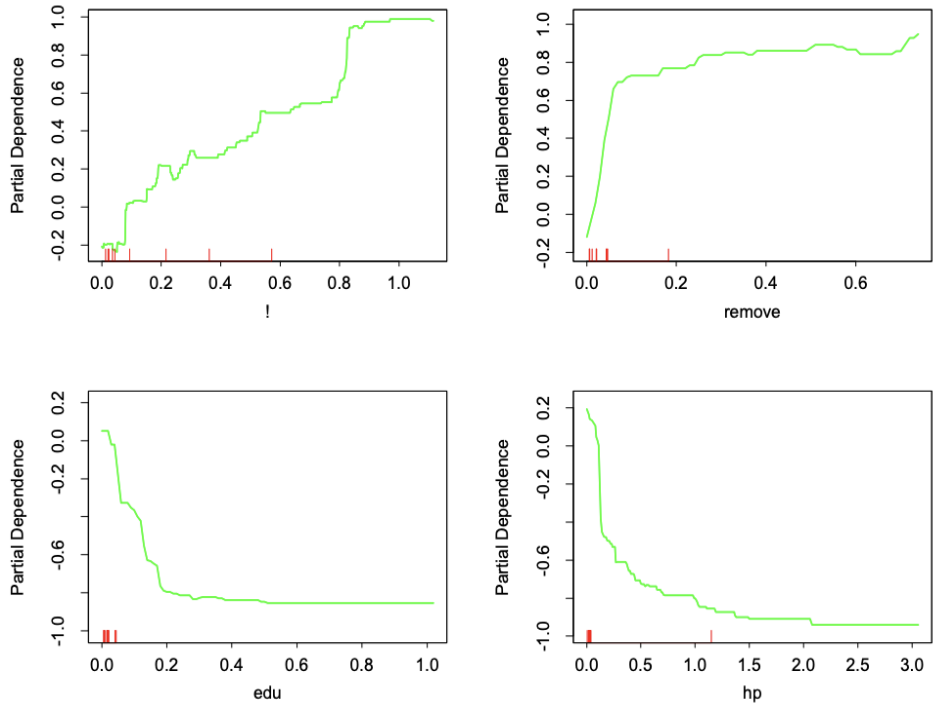


FIGURE 10.7. *Partial dependence of log-odds of* `spam` *on four important predictors. The red ticks at the base of the plots are deciles of the input variable.*

例：スパムデータ

- これらの変数における相互作用の可能性
- `hp` の頻度が低下するにつれて、 `!` との関数的関係が強くなっている

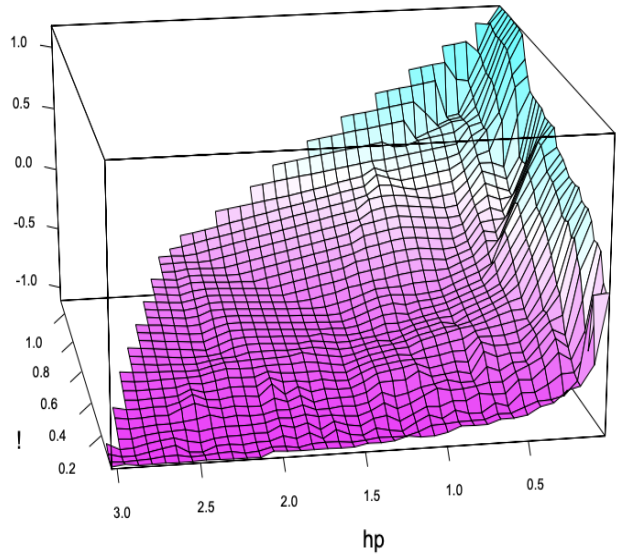


FIGURE 10.8. *Partial dependence of the log-odds of* `spam` *vs.* `email` *as a function of joint frequencies of* `hp` *and the character* `!`.