

機械学習特論 第10章 Boosting

2023/12/20

片岡華瑠

はじめに

10.1 Weak Learnability / **弱学習性**

10.2 AdaBoost

10.3 Linear Combinations of Base Hypotheses / **基本仮説の線形結合**

10.4 AdaBoost for Face Recognition / **顔認識のための AdaBoost**

まとめ

はじめに

はじめに

Boostingは、「2つの主要な問題」を解決するために、線形予測器の一般化を用いる。

1. バイアス-複雑性のトレードオフ (第5章)
2. 学習の計算複雑性 (第8章)

Boostingにより、

- ・ 学習器は、スムーズなトレードオフの制御が可能になる
- ・ 弱い仮説を集約し、より大きく、学習が困難なクラスに対して
徐々に良い予測器を近似するためのツールを提供

この章では、実用的なBoostingアルゴリズムの「AdaBoost」を説明・分析する。

10.1 「Weak Learnability」 / 弱学習可能性

10.1 弱学習可能性

下の2つは、計算的な側面を無視している。

- PAC学習 (第3章)
- 学習理論の基本定理 (第6章 定理6.8)

しかし、「計算の複雑性」と「精度の要件」は引き換え可能かもしれない。

10.1 弱学習可能性（続き）

つまり、下の2つを得た時、

- 分布 \mathcal{D}
- ターゲットラベリング関数 f

下の2つを満たす、学習アルゴリズムが存在するかもしれない。

- 損失がランダムな推測よりも少し優れている
- 効率的に計算可能

定理10.1 「 γ -Weak-Learnability」 / γ -弱学習可能性

定義1

- $\delta \in (0, 1)$
- \mathcal{X} 上の分布 \mathcal{D}
- ラベリング関数 $f: \mathcal{X} \rightarrow \{\pm 1\}$
- 関数 $m_{\mathcal{H}}$ は、 $\mathcal{H}, \mathcal{D}, f$ に関して、実現可能仮定が成り立つ

$m_{\mathcal{H}}: (0, 1)^2 \rightarrow \mathbb{N}$ が存在 \Rightarrow 学習アルゴリズムは γ - 弱学習器

定理10.1 γ -弱学習可能性（続き）

定義1（続き）

以下を満たす、 $m \geq m_{\mathcal{H}}$ i.i.d. examplesを考える。

- \mathcal{D} により生成
- f によりラベル付け

これに対して、学習アルゴリズムを実行

⇓

少なくとも $1-\delta$ の確率で、以下となる仮説 h を返す

- $L_{(D,f)}(h) \leq \frac{1}{2} - \gamma$

定理10.1 γ -弱学習可能性（続き）

定義2

仮説クラス \mathcal{H} が γ -弱学習可能 \Rightarrow そのクラスに対する γ -弱学習器が存在

定理10.1 γ -弱学習可能性（続き）

定義10.1はPAC学習の定義とほぼ同じ。

重要な違いの1つは、「誤差率が $\frac{1}{2} - \gamma$ 以下である仮説の出力で十分」な点。

言い方を変えると、「誤差率がランダムよりも優れていれば十分」な点。

この定義より、効率的な弱学習器を比較的簡単に見つける事が可能かもしれない。

定理6.8 における γ -弱学習可能性

定理6.8 学習の基本定理（第6章）

- \mathcal{H} がVC次元 d を持つ
- C_1 は定数

PAC学習 \mathcal{H} の サンプル複雑性は、

$$m_{\mathcal{H}}(\epsilon, \delta) \geq C_1 \frac{d + \log(1/\delta)}{\epsilon}$$

定理6.8 における γ -弱学習可能性（続き）

$\epsilon = \frac{1}{2} - \gamma$ の時を考える。

- 計算複雑性を考慮しない ($d = \infty$) 時、
 - ・ \mathcal{H} は γ -弱学習可能ではない（PAC学習と同程度に困難）
- 計算複雑性を考慮する時、
 - ・ 弱学習の要件を満たすアルゴリズムが存在する可能性がある
 - ・ 実装可能な可能性がある

弱学習アルゴリズム

単純な仮説クラス B に、ERMを適用する弱学習アルゴリズムを考える。

これが機能するには、以下の2つの要件を満たす必要がある。

- ERM_B は効率的に実装可能
- 任意の ERM_B 仮説の誤差は、最大 $\frac{1}{2} - \gamma$
 - ・ \mathcal{H} からのいくつかの仮説によりラベル付けされた全サンプルにおいて

次に、基本仮説クラス B を使用し、クラス \mathcal{H} の効率的な弱学習可能性を示す例を挙げる。

例10.1

(未完成)

10.1.1 「Efficient Implementation of ERM for Decision Stumps」 / 決定株のためのERMの効率的な実装

- $\mathcal{X} = \mathbb{R}^d$
- \mathbb{R}^d 上の基本仮説クラス
- $b = 1$ と仮定 ($\text{sign}(\theta - x_i)$ 形式の全仮説を考える)

$$\mathcal{H}_{DS} = \{x \mapsto \text{sign}(\theta - x_i) \cdot b : \theta \in \mathbb{R}, i \in [d], b \in \{\pm 1\}\}$$

トレーニングセット $\mathcal{S} = ((\mathbf{x}_1, y_1))$

次に、以下の2つを示す。

- 決定株 $L_{\mathcal{S}}(h)$ の最小化方法を実装する方法
- \mathcal{S} 上の分布に関するリスク関数を最小化する方法

S 上の分布に関するリスク関数を最小化する方法

- 具体的に、 \mathbf{D} は、 \mathbf{R}^m 上の確率ベクトル
 - ・ \mathbf{D} の全要素が非負
 - ・ $\sum_i \mathbf{D}_i = 1$
- 後述する弱い学習者は、
 - ・ \mathbf{D} と S を受け取る
 - ・ \mathbf{D} に対するリスクを最小化する決定スタンプ $h : X \rightarrow Y$ を出力

$$L_{\mathbf{D}}(h) = \sum_{i=1}^m D_i \cdot \mathbb{1}_{[h(\mathbf{x}_i) \neq y_i]}.$$

注意 $\mathbf{D} = (\frac{1}{m}, \dots, \frac{1}{m}) \Rightarrow L_{\mathbf{D}}(h) = L_S(h)$

S 上の分布に関するリスク関数を最小化する方法（続き）

各決定株は、以下の2つによりパラメータ化

- インデックス $j \in [d]$
- 閾値 θ

\Rightarrow 「 $L_D(h)$ の最小化」と下の問題の解決は同じ

$$\min_{j \in [d]} \min_{\theta \in \mathbb{R}} \left(\sum_{i: y_i = 1} D_i \mathbb{1}_{[x_{i,j} > \theta]} + \sum_{i: y_i = -1} D_i \mathbb{1}_{[x_{i,j} \leq \theta]} \right) \quad (10.1)$$

S 上の分布に関するリスク関数を最小化する方法（続き）

- $j \in [d]$ の固定
- $x_{1,j} < x_{2,j} < \dots < x_{m,j}$ となるようにexamplesをソート
- $\Theta_j = \left\{ \frac{x_{i,j} + x_{i+1,j}}{2} : i \in [m-1] \right\} \cup \{(x_{1,j} - 1), (x_{m,j} + 1)\}$ と定義

注意 $\theta' \in \Theta_j$ が存在

- 任意の $\theta \in \mathbb{R}$
- サンプル S に対して、閾値 θ と同じ予測をもたらす

$\Rightarrow \theta \in \Theta_j$ 上で最小化可能 ($\theta \in \mathbb{R}$ 上の代わりに)

10.1.1 決定株のためのERMの効率的な実装（続き）

以下により、に効率的な手順が得られる

- 式(10.1)の目的値を最小にする $j \in [d]$ と $\theta \in \Theta_j$ を選ぶ

実行時間は、 $O(dm^2)$

- 任意の j と $\theta \in \Theta_j$

10.1.1 決定株のためのERMの効率的な実装（続き）

次に、実行時間 $O(dm)$ で目的関数の最小化を行うシンプルな方法を示す

以下において、 $\theta \in (x_{i,j}, x_{i+1,j})$ を考える。

- [仮定] $\theta \in (x_{i-1,j}, x_{i,j})$ の目的関数の計算をした
- 目的関数の値は、 $F(\theta)$

\Downarrow

$$F(\theta') = F(\theta) - D_i \mathbb{1}_{[y_i=1]} + D_i \mathbb{1}_{[y_i=-1]} = F(\theta) - y_i D_i.$$

10.1.1 決定株のためのERMの効率的な実装（続き）

- 以前の閾値 θ での目的関数が与えられた場合

一定時間で θ' の目的関数を計算可能

⇒ 以下の流れで実現

1. 事前処理ステップで、各座標に関してexamplesをソート
2. 最小化問題を $O(dm)$ で実行可能

- 擬似コードあり

10.2 「AdaBoost」

10.2 AdaBoost

- AdaBoostとは、

「弱学習器にアクセスし、低い経験損失を持つ仮説を見つけるアルゴリズム」

- 入力 : $S = (\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)$
- 各 i で、 $y_i = f(\mathbf{x}_i)$ (f はラベリング関数)

Boostingは、連続的なラウンドのシーケンスで進行

10.2 AdaBoost (続き)

```
AdaBoost

input:
  training set  $S = (\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)$ 
  weak learner WL
  number of rounds  $T$ 
initialize  $\mathbf{D}^{(1)} = (\frac{1}{m}, \dots, \frac{1}{m})$ .
for  $t = 1, \dots, T$ :
  invoke weak learner  $h_t = \text{WL}(\mathbf{D}^{(t)}, S)$ 
  compute  $\epsilon_t = \sum_{i=1}^m D_i^{(t)} \mathbb{1}_{[y_i \neq h_t(\mathbf{x}_i)]}$ 
  let  $w_t = \frac{1}{2} \log \left( \frac{1}{\epsilon_t} - 1 \right)$ 
  update  $D_i^{(t+1)} = \frac{D_i^{(t)} \exp(-w_t y_i h_t(\mathbf{x}_i))}{\sum_{j=1}^m D_j^{(t)} \exp(-w_t y_j h_t(\mathbf{x}_j))}$  for all  $i = 1, \dots, m$ 
output the hypothesis  $h_s(\mathbf{x}) = \text{sign} \left( \sum_{t=1}^T w_t h_t(\mathbf{x}) \right)$ .
```

(1) S 内の例に対する分布 $\mathbf{D}(t)$ を定義

- T 回ループ

(2) 弱い仮説 h_t を返す

(3) ϵ の計算

(4) w_t の計算

(5) 分布 $\mathbf{D}(t)$ の更新

10.2 AdaBoost（続き）

○ **Dを定義）** $\mathbf{D}^{(t)} \in \mathbb{R}_+^m$, $\sum_{i=1}^m D_i^{(t)} = 1$ (1)

○ **[仮定] 弱学習器は、弱い仮説 h_t を返す** (2)

$$\epsilon_t \stackrel{\text{def}}{=} L_{\mathbf{D}^{(t)}}(h_t) \stackrel{\text{def}}{=} \sum_{i=1}^m D_i^{(t)} \mathbb{1}_{[h_t(\mathbf{x}_i) \neq y_i]} \quad (3)$$

これに関して、

- 損失は、最大 $\frac{1}{2} - \gamma$ （ランダムより良い）
- 弱学習器の失敗確率は、最大 δ

10.2 AdaBoost（続き）

○ h_t に対して重み割り当てを行う (4)

$$w_t = \frac{1}{2} \log \left(\frac{1}{\epsilon_t} - 1 \right)$$

- h_t の重みは、 h_t の損失に反比例

10.2 AdaBoost（続き）

ラウンドの終わりに分布 D を更新 (5)

- h_t が誤った結果を出すexamplesに高い確率質量を与える
- h_t が正しい結果を出すexamplesに低い確率質量を与える

この分布の更新により、

弱学習器は次のラウンドで問題のあるexamplesに焦点を当てるようになると考えられる。

AdaBoostの出力は、全ての弱い仮説における重み付きの総和に基づく「強い」分類器

定理 10.2

- S はトレーニングセット
- [仮定] AdaBoostの各ループ時、弱学習器が $\epsilon_t \leq \frac{1}{2} - \gamma$ を満たす仮説を返す

AdaBoostの出力仮説の訓練誤差

$$L_S(h_S) = \frac{1}{m} \sum_{i=1}^m \mathbb{1}_{[h_S(x_i) \neq y_i]} \leq \exp(-2\gamma^2 T).$$

定理10.2より、

出力仮説の訓練損失は、Boostingのラウンド数 T に伴い、指数関数的な速度で減少

定理 10.2の証明

- 各 t に対して、 $f_t = \sum_{p \leq t} w_p h_p$
よって、AdaBoostの出力 = f_T
(全ての弱い仮説における重み付きの総和に基づく「強い」分類器)
- 加えて、

$$Z_t = \frac{1}{m} \sum_{i=1}^m e^{-y_i f_t(x_i)}.$$

注意 任意の仮説に対して、 $\mathbb{1}_{[h(x) \neq y]} \leq e^{-yh(x)}$

$$\Rightarrow L_S(f_T) \leq Z_T$$

$$\Rightarrow Z_T \leq e^{-2\gamma^2 T} \text{ が示せば十分}$$

定理 10.2の証明（続き）

- Z_T の上界を求める
 - $Z_0 = 1$ ($f_0 \equiv 0$ より)

$$Z_T = \frac{Z_T}{Z_0} = \frac{Z_T}{Z_{T-1}} \cdot \frac{Z_{T-1}}{Z_{T-2}} \cdots \frac{Z_2}{Z_1} \cdot \frac{Z_1}{Z_0} \quad (10.2)$$

- 任意のラウンド t で、以下が示せれば十分

$$\frac{Z_{t+1}}{Z_t} \leq e^{-2\gamma^2} \quad (10.3)$$

定理 10.2の証明 (続き)

- 単純な帰納法を使用
⇒ 任意の t, i で

$$D_i^{(t+1)} = \frac{e^{-y_i f_t(x_i)}}{\sum_{j=1}^m e^{-y_j f_t(x_j)}}$$

したがって、

$$\frac{Z_{t+1}}{Z_t} = \frac{\sum_{i=1}^m e^{-y_i f_{t+1}(x_i)}}{\sum_{j=1}^m e^{-y_j f_t(x_j)}}$$

$f_t = \sum_{p \leq t} w_p h_p$ より, $f_{t+1}(x_i) = f_t(x_i) + w_{t+1} h_{t+1}(x_i)$ のため,

$$= \frac{\sum_{i=1}^m e^{-y_i f_t(x_i)} e^{-y_i w_{t+1} h_{t+1}(x_i)}}{\sum_{j=1}^m e^{-y_j f_t(x_j)}}$$

定理 10.2の証明（続き）

$$\begin{aligned} &= \sum_{i=1}^m D_i^{(t+1)} e^{-y_i w_{t+1} h_{t+1}(x_i)} \\ &= e^{-w_{t+1}} \sum_{i: y_i h_{t+1}(x_i)=1} D_i^{(t+1)} + e^{w_{t+1}} \sum_{i: y_i h_{t+1}(x_i)=-1} D_i^{(t+1)} \\ &= e^{-w_{t+1}} (1 - \epsilon_{t+1}) + e^{w_{t+1}} \epsilon_{t+1} \\ &= \frac{1}{\sqrt{1/\epsilon_{t+1} - 1}} (1 - \epsilon_{t+1}) + \sqrt{\frac{1}{\epsilon_{t+1}} - 1} (\epsilon_{t+1}) \\ &= \sqrt{\frac{\epsilon_{t+1}}{1 - \epsilon_{t+1}}} (1 - \epsilon_{t+1}) + \sqrt{\frac{1 - \epsilon_{t+1}}{\epsilon_{t+1}}} (\epsilon_{t+1}) \\ &= 2\sqrt{\epsilon_{t+1}(1 - \epsilon_{t+1})}. \end{aligned}$$

定理 10.2の証明（続き）

よって、

$$\frac{Z_{t+1}}{Z_t} = 2\sqrt{\epsilon_{t+1}(1 - \epsilon_{t+1})}$$

- 仮定より, $\epsilon_{t+1} \leq \frac{1}{2} - \gamma$
- $g(a) = a(1 - a)$ は, $[0, \frac{1}{2}]$ において単調増加
⇒ ルートの中が同様に単調増加

$$2\sqrt{\epsilon_{t+1}(1 - \epsilon_{t+1})} \leq 2\sqrt{(\frac{1}{2} - \gamma)(\frac{1}{2} + \gamma)} = \sqrt{1 - 4\gamma^2}$$

定理 10.2の証明（続き）

- $1 - a \leq e^{-a}$ より、

$$\sqrt{1 - 4\gamma^2} \leq e^{-4\gamma^2/2} = e^{-2\gamma^2}$$

- まとめると、

- $\frac{Z_{t+1}}{Z_t} = 2\sqrt{\epsilon_{t+1}(1 - \epsilon_{t+1})}$
- $2\sqrt{\epsilon_{t+1}(1 - \epsilon_{t+1})} \leq \sqrt{1 - 4\gamma^2}$
- $\sqrt{1 - 4\gamma^2} \leq e^{-2\gamma^2}$

⇓

- 先ほどの方程式(10.3)が成立

$$\frac{Z_{t+1}}{Z_t} \leq e^{-2\gamma^2} \quad (10.3)$$

[証明終了]

備考 10.2

- AdaBoostは多くの場合、弱学習器の失敗確率を0として実装可能
- 定理10.2は、AdaBoostによって構築された仮説の経験リスクが、 T の増加につれてゼロに収束することを示している
しかし、本当に気にかけているのは、出力仮説の真のリスク



次のセクションでは、AdaBoostの出力の真のリスクが、
経験リスクから大きく逸脱しないであろう事を示す

10.3 「Linear Combinations of Base Hypotheses」 / 基本仮説の線形結合

10.3 基本仮説の線形結合

- 弱学習器の構築において良く知られた方法は、
基本仮説クラスに関してERMルールの適用する方法
- Boostingは、弱い仮説の予測を基にした半空間の組成物を出力する

↓

AdaBoostの出力は、基本仮説クラス B を与えると、以下のクラスのメンバーになる

$$L(B, T) = \left\{ x \mapsto \text{sign} \left(\sum_{t=1}^T w_t h_t(x) \right) : \mathbf{w} \in \mathbb{R}^T, \forall t, h_t \in B \right\}$$

10.3 基本仮説の線形結合（続き）

- このセクションでは、
BおよびTのVC次元に関連して $L(B, T)$ のVC次元を制約することにより、
 $L(B, T)$ の推定誤差を分析する
- 対数的要因を除く事で、
 $L(B, T)$ のVC次元が、BのVC次元のT倍に制約される事が確認できるであろう
 - AdaBoostの推定誤差は、Tに比例して増加
 - AdaBoostの経験リスクは、Tとともに減少

↓

AdaBoostのパラメータTは、バイアスと複雑性のトレードオフを制御可能にする

10.3 基本仮説の線形結合（続き）

- $L(B, T)$ の表現力が、 T とともにどのような変化をするかを示すため、単純な例で考える
- $\mathcal{X} = \mathbb{R}$
- [仮定] 基本クラスは、決定株

$$\mathcal{H}_{DS1} = \{x \mapsto \text{sign}(x - \theta) \cdot b : \theta \in \mathbb{R}, b \in \{\pm 1\}\}.$$

注意 1次元の場合、 \mathcal{H}_{DS1} は \mathbb{R} 上の半空間と等価

10.3 基本仮説の線形結合（続き）

- \mathcal{H} は、（直線上の半空間と比較して）かなり複雑な区分定数関数
- g_r は、最大 r 個のピースを持つ区分定数関数
⇒ 閾値 $-\infty = \theta_0 < \theta_1 < \theta_2 < \dots < \theta_r = \infty$

$$g_r(x) = \sum_{i=1}^r \alpha_i \mathbb{1}_{[x \in (\theta_{i-1}, \theta_i]]} \quad \forall i, \alpha_i \in \{\pm 1\}.$$

10.3 基本仮説の線形結合（続き）

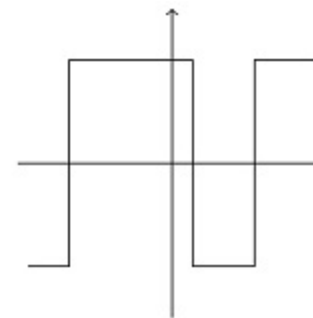
- \mathcal{G}_r は、最大 r 個のピースを持つ全ての区分定数分類器のクラスとする
次に、以下を示す。

$$\mathcal{G}_T \subseteq L(\mathcal{H}_{DS1}, T)$$

（ T 個の決定株に関する半空間のクラスは、
最大 T 個のピースを持つ全ての区分定数分類器を生成する）

10.3 基本仮説の線形結合（続き）

- $\alpha_t = (-1)^t$ である、任意の $g \in \mathcal{G}_T$ を考える
 - x が $(\theta_{t-1}, \theta_t]$ $\Rightarrow g(x) = (-1)^t$



$$h(x) = \text{sign} \left(\sum_{t=1}^T w_t \text{sign}(x - \theta_{t-1}) \right)$$

- この時、
- $w_1 = 0.5$
 - $t > 1$ で、 $w_t = (-1)^t$
 - $L(\mathcal{H}_{DS1}, T)$ に属す
 - g と等しい

10.3 基本仮説の線形結合（続き）

- $L(\mathcal{H}_{DS1}, T)$ は、 \mathbb{R} の $T+1$ のインスタンスの集合を分割可能
 - $\Rightarrow L(\mathcal{H}_{DS1}, T)$ のVC次元は、 $T+1$ 以上
 - $\Rightarrow T$ は、バイアス-複雑性のトレードオフを制御できるパラメータ
- T を拡大するとより表現力のある仮説クラスが得られるが、一方で推定誤差が増加する可能性がある

10.3.1 $L(B, T)$ のVC次元

- このサブセクションでは、
任意の基本クラス B に対して、 $L(B, T)$ のVC次元を形式的に上界とする

補題 10.3

- [仮定] $T, \text{VCdim}(B) > 3$
$$\text{VCdim}(L(B, T)) \leq T(\text{VCdim}(B) + 1)(3 \log(T(\text{VCdim}(B) + 1)) + 2)$$

この補題では、 $L(B, T)$ のVC次元は $\tilde{O}(\text{VCdim}(B)T)$ を上界とする事を示している

補題10.3の証明

- $d = \text{VCdim}(B)$
- $C = \{x_1, \dots, x_m\}$ は、 $L(B, T)$ によって細分される集合
- C を $L(B, T)$ によってラベリング
- サウアーの補題より、
 C 上の B で、導かれる異なるラベリングは最大で $(\frac{em}{d})^d$ 個だけ
 \Rightarrow • 最大で $(\frac{em}{d})^d$ 個の異なる仮説の出力から、 T 個の仮説を選ぶ必要がある
 • これを行う方法は最大で $(\frac{em}{d})^{dT}$ 通り

補題10.3の証明（続き）

- 次に、各選択肢に対して、最大で $(\frac{em}{T})^T$ 個の二分法を生成する線形予測器を適用
⇒
 - ・ 二分法の総数は、以下の不等式による上界で構築できる

$$\left(\frac{em}{d}\right)^{dT} \left(\frac{em}{T}\right)^T \leq m^{(d+1)T}$$

- ・ [仮定] $d, T \geq 3$

補題10.3の証明（続き）

- C が細分されていると仮定 $\Rightarrow 2^m$ 以上

$$2^m \leq m^{(d+1)T}$$

したがって、

$$m \leq \log(m) \frac{(d+1)T}{\log(2)}$$

- Chapter Aの補題 A.1より、上記が成り立つための必要条件は以下となる
[補題A.1] $x < \log(x)$ が成り立つための必要条件は、 $x < 2a \log(a)$

$$m \leq 2 \frac{(d+1)T}{\log(2)} \log \frac{(d+1)T}{\log(2)} \leq (d+1)T(3 \log((d+1)T) + 2)$$

[証明終了]

L(B, T)のVC次元 (続き)

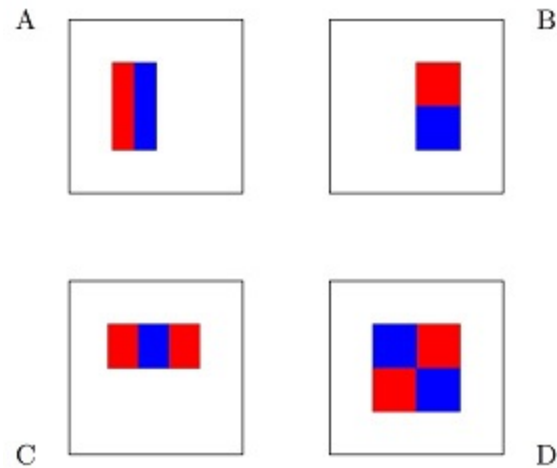


図10.1 :

- A, Bにおける関数 g の値は、2つの長方形領域内のピクセルの合計の差
- Cにおける関数 g の値は、
「中央の長方形内の合計」から「外側の2つの長方形内の合計」を引いた値
- Dにおける関数 g の値は、長方形の対角線のペア間の差

10.4 「AdaBoost for Face Recognition」 / 顔認識のための AdaBoost

10.4 顔認識のための AdaBoost

- 顔認識のタスクのために、ViolaとJonesによって提案された基本仮説について検討
 - ・ インスタンス空間は画像であり、画素の灰色値の行列として表現される
(サイズが 24×24 の実数値行列の集合)
- ゴールは、以下の入出力を行う分類器の学習
 - ・ 入力 : 画像
 - ・ 出力 : 画像が人間の顔であるか否か

$$h : \chi \rightarrow \{\pm 1\}$$

10.4 顔認識のための AdaBoost（続き）

- 基本クラス内の各仮説は、 $h(x) = f(g(x))$
 - ・ f は、決定株仮説
 - ・ $g : \mathbb{R}^{24,24} \rightarrow \mathbb{R}$
- 各関数 g は、以下によりパラメータ化されている
 - ・ 軸に並行な長方形 R （画像のサイズが $24 \times 24 \Rightarrow$ 最大 24^4 ）
 - ・ $t \in A, B, C, D$ （マスクのタイプ）

10.4 顔認識のための AdaBoost（続き）

- g の計算
 1. マスク t を長方形 R に合わせて伸縮
 2. 「赤い長方形のピクセルの総和」と「青い長方形のピクセルの総和」を計算
 3. その2つの総和の差を計算
- 関数 g の数は最大で $24^4 \cdot 4$
 - ⇒ ・ 以下により、基本的な仮説クラスの弱学習器を実装可能
 1. 各画像で g の可能な全ての出力を計算
 2. 10.3.1で説明した決定木の弱学習器を適用

10.4 顔認識のための AdaBoost (続き)

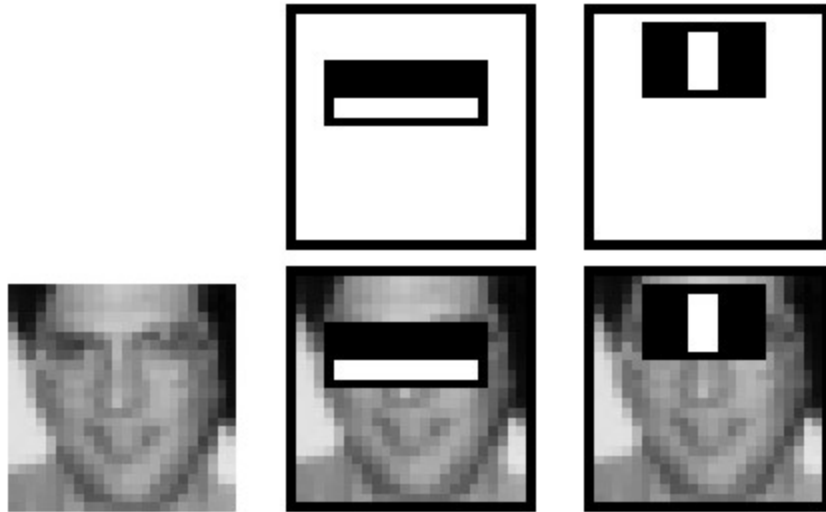


図 10.2

- 最初の特徴は、目の領域と上頬の領域との間の明度の差を測定している
この特徴は、目の領域が通常頬よりも暗いことが観察される事を活用している
- 2つ目の特徴は、目の領域の明度を鼻梁にわたる明度と比較している

まとめ

- ブースティングは、弱学習器の精度を高める手法
- この章では、AdaBoostアルゴリズムについて説明
- AdaBoostの T 回のループ後、 $L(B, T)$ クラスから仮説を返すことを示した
- パラメーター T が、近似と推定誤差のトレードオフを制御する方法を示した
- 次の章は、データに基づき、 T 等のパラメータを調整する方法に関して説明