情報生命科学基礎Ⅰ
情報生命科学演習

7月10／17日

森下

演習内容

- ７月１０日の講義
  AdaBoost アルゴリズムの説明

- ７月１０日の演習
  プログラム例を紹介

- ７月１７日
  Jones and Pevzner
  An Introduction to Bioinformatics Algorithms
  Chapter 4　Exhaustive Search

- レポートの〆切　７月３１日
  あて先　斉藤助教　　leo@cb.k.u-tokyo.ac.jp

AdaBoost の応用例

Marcel Dettling and Peter Bühlmann
Boosting for tumor classification with gene expression data
Bioinformatics, Jun 2003; 19: 1061 - 1069.

Jinyan Li, Huiqing Liu, See-Kiong Ng, and Limsoon Wong
Discovery of significant rules for classifying cancer diagnosis data
Bioinformatics, Sep 2003; 19: 93 - 102.

Manuel Middendorf, Anshul Kundaje, Chris Wiggins, Yoav Freund, and Christina Leslie
Predicting genetic regulatory response using classification
Bioinformatics, Aug 2004; 20: i232 - i240.

Pål Sætrom
Predicting the efficacy of short oligonucleotides in antisense and RNAi experiments with boosted genetic programming
Bioinformatics, Advance Access published on June 16, 2004; doi: 10.1093/bioinformatics/bth364.

三人寄れば文殊の知恵

Boosting とは正答率の低いクラス分類器を組合せて、
高い正答率を示すクラス分類器を構成する技法。

参考文献

Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119-139, 1997.

訓練用データ

	T1	T2	T3	T4	目標属性	
$\vec{x}_1$	1	0	1	1	1	y_1
	1	0	1	1	1	
	1	1	1	1	1	
	1	1	1	0	0	
	1	0	1	0	0	
	1	1	0	1	0	
	1	0	0	1	0	
	1	1	0	1	0	
	0	1	0	1	1	
	0	0	1	1	1	
	0	1	0	1	1	
	0	1	0	1	1	
	0	0	0	1	1	
	0	0	1	0	0	
	0	1	0	0	0	
	0	0	1	0	0	

$$(\vec{x}_1, y_1)$$

$$\vdots$$

$$(\vec{x}_N, y_N)$$

AdaBoost の基本的考え方

- 初期状態では各レコードに均等な重みを割当てる
- 次のステップを繰返す

1. ランダムに推測するよりは正答率の高い（50%を超える）クラス分類器を生成
2. 目標属性の値の予測を誤ったレコードの重みを相対的に上げる（予測が困難であることを覚えておく）

註：参考論文中ではクラス分類器(classifier)のことを仮説(hypothesis)と呼んでいる

クラス分類器

T1	T2	T3	T4	目標	重み	if T1=1 then Ob=0 else Ob=1	新しい重み
1	0	1	1	1	■	0	■
1	0	1	1	1	■	0	■
1	1	1	1	1	■	0	■
1	1	1	0	0	■	0	■
1	0	1	0	0	■	0	■
1	1	0	1	0	■	0	■
1	0	0	1	0	■	0	■
1	1	0	1	0	■	0	■

■ の大きさが重みを表現

T1	T2	T3	T4	目標	重み	新たな クラス分類器 if T3=1 then Ob=1 else Ob=0	新しい 重み
1	0	1	1	1		1	
1	0	1	1	1		1	
1	1	1	1	1		1	
1	1	1	0	0		1	
1	0	1	0	0		1	
1	1	0	1	0		0	
1	0	0	1	0		0	
1	1	0	1	0		0	

T1	T2	T3	T4	目標	重み	新たな クラス分類器 if T4=1 then Ob=1 else Ob=0	新たな 重み
1	0	1	1	1	■	1	■
1	0	1	1	1	■	1	■
1	1	1	1	1	■	1	■
1	1	1	0	0	■	0	■
1	0	1	0	0	■	0	■
1	1	0	1	0	■	1	■
1	0	0	1	0	■	1	■
1	1	0	1	0	■	1	■

T1	T2	T3	T4	Ob	クラス分類器 if T1=1 then Ob=0 else Ob=1	 if T3=1 then Ob=1 else Ob=0	 if T4=1 then Ob=1 else Ob=0	単純な 多数決
1	0	1	1	1	0	1	1	1
1	0	1	1	1	0	1	1	1
1	1	1	1	1	0	1	1	1
1	1	1	0	0	0	1	0	0
1	0	1	0	0	0	1	0	0
1	1	0	1	0	0	0	1	0
1	0	0	1	0	0	0	1	0
1	1	0	1	0	0	0	1	0

AdaBoost は重みを使った多数決

AdaBoost　アルゴリズム

入力

訓練データ　　$(\vec{x}_1, y_1), \ldots, (\vec{x}_N, y_N)$

初期の重み　　$w_i^1 = D(i) = 1/N$　　for each i =1,2, …, N

WeakLearn：　エラー率が0.5未満の
クラス分類器を常に出力する学習アルゴリズム

T: 最終的に使うクラス分類器の個数

各 $t = 1, ..., T,$ について以下のステップを繰返す:

1: 各データの重みを正規化し、分布 p_i^t を計算

$$p_i^t = w_i^t \Big/ \sum_{i=1}^{N} w_i^t$$

2: **WeakLearn** を呼び出し次の条件を満たすクラス分類器 h_t を生成

$$\varepsilon_t = \sum_{i=1}^{N} p_i^t \left| h_t(\vec{x}_i) - y_i \right| < 1/2$$

$$\left| h_t(\vec{x}_i) - y_i \right| = \begin{cases} 0 & h_t \text{が} \vec{x}_i \text{の正解を返す場合} \\ 1 & \text{otherwise} \end{cases}$$

3: 重みを更新　重みは正解だと軽くなり、不正解だとそのまま

$$\beta_t = \varepsilon_t / (1 - \varepsilon_t) \qquad \beta_t < 1 \qquad w_i^{t+1} = w_i^t \beta_t^{1 - \left| h_t(\vec{x}_i) - y_i \right|}$$

$\varepsilon_t \to 0, \quad \beta_t \to 0 \qquad \varepsilon_t \to 1/2, \quad \beta_t \to 1$

出力: 最終のクラス分類器 h_f (h_t の重みつき多数決)

$$h_f(\vec{x}) = \begin{cases} 1 & \text{if} \quad \sum_{t=1}^{T}(-\log\beta_t)h_t(\vec{x}) \geq \sum_{t=1}^{T}(-\log\beta_t)\dfrac{1}{2} \quad \cdots(*) \\ 0 & \text{otherwise} \end{cases}$$

$\varepsilon_t \to 0, \quad \beta_t \to 0, \quad (-\log\beta_t) \to +\infty \qquad \varepsilon_t \to 1/2, \quad \beta_t \to 1, \quad (-\log\beta_t) \to 0$

エラー率が低い予測を尊重する

定義 $\varepsilon = \sum_{\{i \mid h_f(\vec{x}_i) \neq y_i\}} D(i)$

最終クラス分類器 h_f の
初期分布に対するエラー率
$D(i) = 1/N$

定理 $\varepsilon \leq \prod_{t=1}^{T} 2\sqrt{\varepsilon_t(1-\varepsilon_t)} = \prod_{t=1}^{T}\sqrt{1-(1-2\varepsilon_t)^2}$

$\varepsilon_t < 1/2, \quad 1-2\varepsilon_t < 1, \quad \sqrt{1-(1-2\varepsilon_t)^2} < 1$

レポートの内容

1. プログラムのソースコード
   サンプルプログラムを与える
   自ら考えて実装した場合は可以上の評価
   サンプルプログラムをそのまま使用した場合は可以下の評価
2. データ構造と手続きの流れの説明
3. サンプルデータ（２値）から計算された分類器を示すこと

4. 重み、エラー率の変動についての解析
5. データ数を 100, 1000 と増加させた例を作って計算時間とエラー率を測定する

6. WeakLearn が新しいクラス分類器を出力できない例
7. WeakLearn が (T1=1) かつ (T2=0) のように2つの属性を使ったクラス分類器を出力できるように拡張する

サンプルデータ（2値）

T1	T2	T3	T4	目標属性
1	0	1	1	1
1	0	1	1	1
1	1	1	1	1
1	1	1	0	0
1	0	1	0	0
1	1	0	1	0
1	0	0	1	0
1	1	0	1	0

T1	T2	T3	T4	目標属性
1	0	1	1	1
1	0	1	1	1
1	1	1	1	1
1	1	1	0	0
1	0	1	0	0
1	1	0	1	0
1	0	0	1	0
1	1	0	1	0
0	1	0	1	1
0	0	1	1	1
0	1	0	1	1
0	1	0	1	1
0	0	0	1	1
0	0	1	0	0
0	1	0	0	0
0	0	1	0	0

Boosting 性の証明

定義　$\varepsilon = \sum_{\{i \mid h_f(\vec{x}_i) \neq y_i\}} D(i)$

最終クラス分類器 h_f の
初期分布に対するエラー率

定理　$\varepsilon \leq \prod_{t=1}^{T} 2\sqrt{\varepsilon_t(1-\varepsilon_t)} = \prod_{t=1}^{T} \sqrt{1-(1-2\varepsilon_t)^2}$

$\varepsilon_t < 1/2, \quad 1-2\varepsilon_t < 1, \quad \sqrt{1-(1-2\varepsilon_t)^2} < 1$

証明のロードマップ

$$\varepsilon\left(\prod_{t=1}^{T}\beta_t\right)^{1/2} \le \sum_{i=1}^{N} w_i^{T+1}$$

補題２　最終回の重みの総和の下限値

$$\le \left(\sum_{i=1}^{N} w_i^{T}\right)\times 2\varepsilon_t$$

補題３　$\sum_{i=1}^{N} w_i^{t+1} \le \left(\sum_{i=1}^{N} w_i^{t}\right)\times 2\varepsilon_t$

重みの総和は $2\varepsilon_t$ 倍以下になる

$$\vdots$$

$$\le \left(\sum_{i=1}^{N} w_i^{1}\right)\prod_{t=1}^{T} 2\varepsilon_t \qquad \sum_{i=1}^{N} w_i^{1} = 1$$

補題3の繰返し適用

$$w_i^{1} = D(i) = 1/N$$

$$\varepsilon \le \prod_{t=1}^{T}\left(2\varepsilon_t \times \beta_t^{-1/2}\right) = \prod_{t=1}^{T} 2\sqrt{\varepsilon_t(1-\varepsilon_t)}$$

$\beta_t = \varepsilon_t/(1-\varepsilon_t)$ に注意

補題 2　$\sum_{i=1}^{N} w_i^{T+1} \geq \varepsilon \left(\prod_{t=1}^{T} \beta_t \right)^{1/2}$

最終回の重みの総和の下限値

$$\sum_{i=1}^{N} w_i^{T+1} \geq \sum_{\{i \mid h_f(\vec{x}_i) \neq y_i\}} w_i^{T+1}$$

最終クラス分類器 h_f が誤るレコードの重みの総和

$$= \sum_{\{i \mid h_f(\vec{x}_i) \neq y_i\}} \left(D(i) \prod_{t=1}^{T} \beta_t^{1-|h_t(\vec{x}_i)-y_i|} \right)$$

$w_i^{t+1} = w_i^t \beta_t^{1-|h_t(\vec{x}_i)-y_i|}$ より

$$\geq \sum_{\{i \mid h_f(\vec{x}_i) \neq y_i\}} \left(D(i) \left(\prod_{t=1}^{T} \beta_t \right)^{1/2} \right)$$

補題 1 より

$$= \left(\sum_{\{i \mid h_f(\vec{x}_i) \neq y_i\}} D(i) \right) \left(\prod_{t=1}^{T} \beta_t \right)^{1/2}$$

最終クラス分類器 h_f が誤るレコードの重みは

$$\left(\beta_1 \cdots \beta_T\right)^{1/2}$$

以上になる（あまり減らない）

$$h_f(\vec{x}) = \begin{cases} 1 & \text{if} \quad \sum_{t=1}^{T}(-\log\beta_t)h_t(\vec{x}) \geq \sum_{t=1}^{T}(-\log\beta_t)\dfrac{1}{2} \quad \cdots(*) \\ 0 & \text{otherwise} \end{cases}$$

補題1　$h_f(\vec{x}_i) \neq y_i$ ならば $\displaystyle\prod_{t=1}^{T}\beta_t^{1-|h_t(\vec{x}_i)-y_i|} \geq \left(\prod_{t=1}^{T}\beta_t\right)^{1/2}$

コメント　最終のクラス分類機で予測が失敗するデータでは重みが十分に減らない

両辺のlog をとった

$\sum_{t=1}^{T}(\log\beta_t)(1-|h_t(\vec{x}_i)-y_i|) \geq \sum_{t=1}^{T}(\log\beta_t)\dfrac{1}{2}$　を証明

$h_f(\vec{x}_i)=1, y_i=0$ ならば：　$\sum_{t=1}^{T}(\log\beta_t)$ を(*)の両辺に加算.

$\sum_{t=1}^{T}(\log\beta_t)(1-h_t(\vec{x}_i)) \geq \sum_{t=1}^{T}(\log\beta_t)\dfrac{1}{2}$

さらに $1-h_t(\vec{x}_i)=1-|h_t(\vec{x}_i)-y_i|$ に注意

$h_f(\vec{x}_i)=0, y_i=1$ ならば：　$\sum_{t=1}^{T}(-\log\beta_t)h_t(\vec{x}_i) < \sum_{t=1}^{T}(-\log\beta_t)\dfrac{1}{2}$

$\sum_{t=1}^{T}(\log\beta_t)h_t(\vec{x}_i) > \sum_{t=1}^{T}(\log\beta_t)\dfrac{1}{2}$

さらに $h_t(\vec{x}_i)=1-|h_t(\vec{x}_i)-y_i|$ に注意

補題3　$\sum_{i=1}^{N} w_i^{t+1}$

$$= \sum_{i=1}^{N} w_i^t \beta_t^{1-|h_t(\vec{x}_i)-y_i|}$$

$$\leq \sum_{i=1}^{N} w_i^t (1-(1-\beta_t)(1-|h_t(\vec{x}_i)-y_i|))$$

$\alpha^\gamma \leq 1-(1-\alpha)\gamma,$
if $\alpha \leq 1$ and $\gamma = 0$ or 1.

$$= \sum_{i=1}^{N} w_i^t - (1-\beta_t)\sum_{i=1}^{N} w_i^t (1-|h_t(\vec{x}_i)-y_i|)$$

正解する重みの和

$$1-\varepsilon_t = \sum_{i=1}^{N} p_i^t - \sum_{i=1}^{N} p_i^t |h_t(\vec{x}_i)-y_i|$$
$$= \sum_{i=1}^{N} p_i^t (1-|h_t(\vec{x}_i)-y_i|)$$
$$= \sum_{i=1}^{N} w_i^t (1-|h_t(\vec{x}_i)-y_i|) \Big/ (\sum_{i=1}^{N} w_i^t)$$

$$= (\sum_{i=1}^{N} w_i^t)(1-(1-\beta_t)(1-\varepsilon_t))$$

$$= (\sum_{i=1}^{N} w_i^t) \bullet 2\varepsilon_t$$

$$1-\beta_t = 1-\frac{\varepsilon_t}{1-\varepsilon_t} = \frac{1-2\varepsilon_t}{1-\varepsilon_t}$$