クラス分類の技法

AdaBoost
K nearest neighbors
Naïve Bayes

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.

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

■　の大きさが重みを表現

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	■

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	■

新たな
クラス分類器

T1	T2	T3	T4	目標	重み	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

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

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	

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

$$\vdots$$

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

各 $t=1,\ldots,T,$ について以下のステップを繰返す:

1: 各レコードの重みを正規化して各レコードの分布 p_i^t を計算

$$p_i^t = \begin{cases} w_i^t \Big/ \sum_{i=1}^{N} w_i^t & \text{if } \sum_{i=1}^{N} w_i^t > 0 \\ 0 & \text{otherwise} \end{cases}$$

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

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

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

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

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

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

AdaBoost の弱点は WeakLearn が条件を満たすクラス分類器を出力できない場合があることである

ひとつもクラス分類器を出力できない訓練データの例

T1	目標属性
0	1
0	0
1	1
1	0

If T1 = 0 then 0 else 1
If T1 = 1 then 0 else 1
エラー率 0.5

T1	T2	目標属性
0	0	1
0	0	0
0	1	1
0	1	0
1	0	1
1	0	0
1	1	1
1	1	0

If T1 = 0 then 0 else 1
If T2 = 0 then 0 else 1
If (T1 = 0) and (T2 = 0) then 0 else 1
If (T1 = 0) and (T2 = 1) then 0 else 1
エラー率 0.5

出力:　最終のクラス分類器 h_f (h_t の重みつき多数決)

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

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

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

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

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

$$\vdots$$

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

$\sum_{i=1}^{N} w_i^{1} = 1$　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

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

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

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

$w_i^{t+1} = w_i^t \beta_t^{1-\left|h_t(\vec{x}_i)-y_i\right|}$ 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
$(\beta_1 \cdots \beta_T)^{1/2}$
以上になる（あまり減らない）

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

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

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

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

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

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

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

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

補題3　$\sum_{i=1}^{N} w_i^{t+1}$

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

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

$$\boxed{\begin{array}{l} \alpha^\gamma \le 1-(1-\alpha)\gamma, \\ \text{if } \alpha \le 1 \text{ and } \gamma = 0 \text{ or } 1. \end{array}}$$

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

正解する重みの和

$$\boxed{\begin{aligned} 1-\varepsilon_t &= \sum_{i=1}^{N} p_i^t - \sum_{i=1}^{N} p_i^t |h_t(\vec{x}_i)-y_i| \\ &= \sum_{i=1}^{N} p_i^t (1-|h_t(\vec{x}_i)-y_i|) \\ &= \sum_{i=1}^{N} w_i^t (1-|h_t(\vec{x}_i)-y_i|) \Big/ \left(\sum_{i=1}^{N} w_i^t\right) \end{aligned}}$$

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

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

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

ACM, Paris Kanellakis Theory and Practice Award

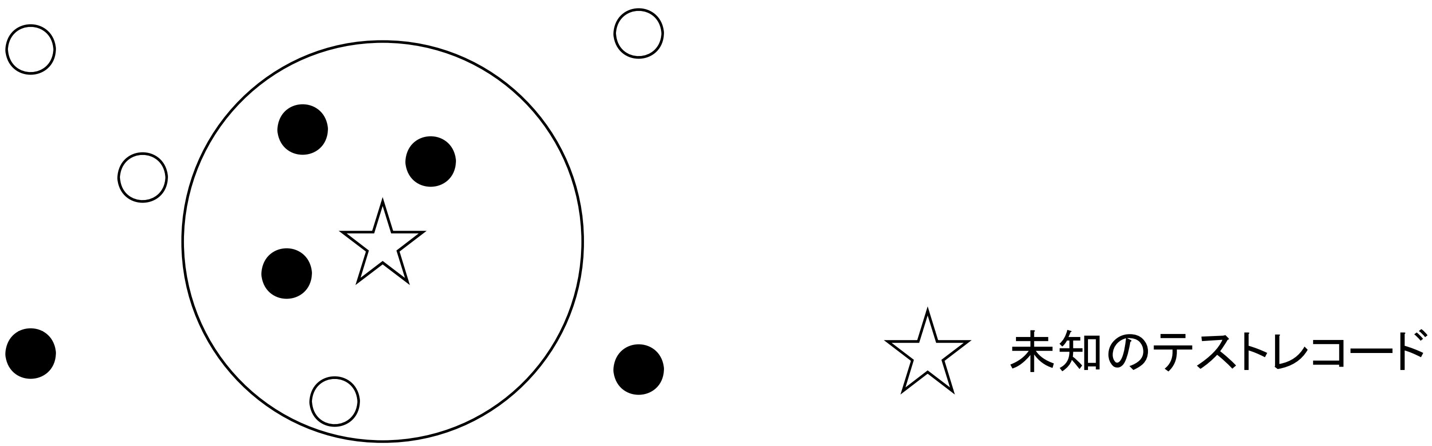
2004
Yoav Freund, *Columbia University*   Robert Schapire, *Princeton University*
For the development of the theory and practice of boosting and its applications to machine learning.

2001
Eugene W. Myers University of Arizona/Celera Genomics/ UCB
For distinguished contributions to the theory of sequence analysis and its application to the sequencing of the human genome and the development of BLAST.

1996
Leonard Adleman, Whitfield Diffie, Martin Hellman, Ralph Merkle, Ronald Rivest, Adi Shamir
Public-Key Cryptography

k nearest neighbors

- 規則を学習しない
- 未知のテストレコードと最も類似した(適切な距離を定義し、
  近い距離にある)k 個の訓練レコードの集合 S を求める
- 未知のテストレコードの目標属性値を S から予測
  例　　２値の場合:　S の目標属性値の多数決
  　　　数値の場合:　　平均値
- レコード間の類似性を適切に定義することが重要で
  ユークリッド距離やハミング距離など
- k の選択も学習能力を左右



ナイーブベイズ分類

条件付確率

$$P(A \mid T) = P(A \,\&\, T) \,/\, P(T)$$

$$P(T \mid A) = P(A \,\&\, T) \,/\, P(A)$$

$$P(A \,\&\, T) = P(A \mid T)\, P(T) = P(T \mid A)\, P(A)$$

ベイズの定理

$$P(A \mid T) = \frac{P(T \mid A) \qquad P(A)}{P(T)}$$

事後確率
（条件TのもとAが観測される確率）

尤度
（Aが観測されたとき、それを
説明する条件 T の尤もらしさ）

事前確率

訓練データ

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

未知のテストデータに対して
クラス分類器　T1=v1 ,…, T4=v4　が目標属性値 w を予測する(事後)確率をベイズの定理より推定

$$P(A=w \mid T1=v1, \ldots, T4=v4) = \frac{P(A=w)\ \ P(T1=v1, \ldots, T4=v4 \mid A=w)}{P(T1=v1, \ldots, T4=v4)}$$

尤度（目標属性Aの値　w が観測されたとき、それを説明する条件 T1=v1 ,…, T4=v4 の尤もらしさを示す）

T1=v1 ,…, T4=v4　は　w の値として 1 と 0 どちらを推定するか？
事後確率の推定　尤度を条件付確率を利用して変形

$P(T1\&T2 \mid A) = P(T1 \mid A)\ P(T2 \mid T1\&\ A)$

$$
\begin{aligned}
& P(T1=v1, T2=v2, T3=v3, T4=v4 \mid A=w) \\
&= P(T1=v1 \mid A=w)\ P(T2=v2, T3=v3, T4=v4 \mid T1=v1, A=w) \\
&= P(T1=v1 \mid A=w) \\
&\quad P(T2=v2 \mid T1=v1, A=w)\ P(T3=v3, T4=v4 \mid T2=v2, T1=v1, A=w) \\
&= P(T1=v1 \mid A=w)\ P(T2=v2 \mid T1=v1, A=w) \\
&\quad P(T3=v3 \mid T2=v2, T1=v1, A=w)\ P(T4=v4 \mid T3=v3, T2=v2, T1=v1, A=w) \\
&\fallingdotseq P(T1=v1 \mid A=w)\ P(T2=v2 \mid A=w)\ P(T3=v3 \mid A=w)\ P(T4=v4 \mid A=w)
\end{aligned}
$$

条件A=w のもと T1=v1 ,…, T4=v4 が互いに独立であることを「ナイーブに」仮定　（ナイーブベイス分類）
独立性 $P(T1=v1\ \&\ T2=v2 \mid A=w) = P(T1=v1 \mid A=w)\ P(T2=v2 \mid A=w)$ と
性質　$P(T1=v1\ \&\ T2=v2 \mid A=w) = P(T1=v1 \mid A=w)\ P(T2=v2 \mid T1=v1\ \&\ A=w)$ から
$P(T2=v2 \mid T1=v1, A=w) = P(T2=v2 \mid A=w)$ が導かれる

訓練データ

T1	T2	T3	T4	目標属性A
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

条件付確率　P(T*i* =*v* | A=*w*)

		目標属性 A	
	w / *v*	1	0
T1	1	3/8	5/8
	0	5/8	3/8
T2	1	4/8	4/8
	0	4/8	4/8
T3	1	4/8	4/8
	0	4/8	4/8
T4	1	8/8	3/8
	0	0/8	5/8

尤度

P(T1=0, T2=0, T3=0, T4=1 | A=1)
= P(T1=0|A=1) P(T2=0|A=1) P(T3=0|A=1) P(T4=1|A=1)
= (5/8) (4/8) (4/8) (8/8)
P(T1=0, T2=0, T3=0, T4=1 | A=0)
= (3/8) (4/8) (4/8) (3/8)

ナイーブベイズ分類器

クラス分類器　T1＝v1 ,…, T4＝v4　が推定する目標属性値 w は

事前確率　×　尤度　＝

P(A＝w) P(T1＝v1 | A＝w) P(T2＝v2 | A＝w) P(T3＝v3 | A＝w) P(T4＝v4 | A＝w)

を最大にする w の値と定義する（右辺の値は訓練データから収集する）

例

P(A＝1) P(T1＝0|A＝1) P(T2＝0|A＝1) P(T3＝0|A＝1) P(T4＝1|A＝1)
= (1/2) (5/8) (4/8) (4/8) (8/8)

> P(A＝0) P(T1＝0|A＝0) P(T2＝0|A＝0) P(T3＝0|A＝0) P(T4＝1|A＝0)
= (1/2) (3/8) (4/8) (4/8) (3/8)

T1＝0, T2＝0, T3＝0, T4＝1 は A＝1　を推定