

# Rodziny klasyfikatorów na przykładzie algorytmów wzmacniających.

## BOOSTING

Iwona Głowacka

19 listopada 2003 roku

### 1 Wprowadzenie

W ogólności analiza dyskryminacyjna polega na budowie reguły klasyfikacyjnej (klasyfikatora) w oparciu o pewien „uczący” zbiór danych. Poniższy tekst ma na celu omówienie problemu klasyfikacji z wykorzystaniem relatywnie małego zbioru danych. W sytuacji takiej powstały klasyfikator może okazać się **niestabilnym** (klasyfikator obciążony z dużą wariancją). Objawia się to dużym błędem klasyfikacji obserwacji należących do próbki testowej, przy dobrej klasyfikacji obserwacji z próbki uczącej. Istnieją pewne metody niwelowania takiej niestabilności, zwane **metodami stabilizacji klasyfikatorów**. Do najbardziej popularnych należą:

1. **bagging**,
2. **boosting**,
3. metoda: **Random Subspace**.

W metodach tych wykorzystuje się **łączenie klasyfikatorów** (rodziny klasyfikatorów) wraz z **metodami głosowań**. Opiszemy to na przykładzie **boostingu** (*boosting - wzmacnianie*).

### 2 Boosting

Boosting jest ogólną metodą, u podstaw której leży idea poprawy dokładności działania dowolnego algorytmu uczącego. Technika ta polega na stosowaniu sekwencji prostych modeli, przy czym każdy kolejny model *przykłada większą „wagę”* do tych obserwacji, które zostały błędnie zaklasyfikowane przez poprzednie modele.

## 2.1 Boosting w przeszłości

Podstawy boostingu powstały podczas prac nad strukturą modelu uczącego zwanego **PAC**. Pierwszymi, którzy zastanawiali się nad możliwością wzmocnienia „słabego” (ang. - weak) algorytmu uczącego, którego wyniki działania są nieco lepsze od losowych (random guessing w modelu PAC), byli *Kearns* i *Valiant*. Wynikiem takiego wzmocnienia powinien być dowolnie dokładny „silny” algorytm uczący. W 1989 r. *Shapire* przedstawił pierwszy algorytm wzmacniający działający w czasie wielomianowym. Rok później *Freund* wprowadził znacznie efektywniejszą wersję boostingu, niestety optymalną tylko w pewnym sensie i obciążoną praktycznymi wadami.

## 2.2 Algorytm AdaBoost - boosting adaptacyjny

Zasadę działania boostingu przedstawimy na przykładzie algorytmu AdaBoost. Algorytm ten został wprowadzony w 1995 r. przez *Freund'a* i *Shapire'a*. Pozwolił on rozwiązać większość praktycznych problemów jakimi były obciążone wcześniejsze wersje boostingu.

### 2.2.1 AdaBoost - założenia

Parametry wejściowe algorytmu AdaBoost to:

- zbiór uczący  $S$  o  $m$  elementach pochodzących z  $K$  klas  
 $S = \{(x_i, y_i), i = 1, 2, \dots, m\}, \quad y_i \in Y = \{1, \dots, K\}$   
 $x_i$  -  $i$  - ta obserwacja  
 $y_i$  - klasa  $i$  - tej obserwacji
- algorytm uczący  
**LEARN** - funkcja tworząca klasyfikator na podstawie zbioru  $S$   
z uwzględnieniem wcześniej nadanych wag
- stała  $L$   
 $L$  - maksymalna liczba iteracji

**Definicja 2.1** Niech  $E$  będzie daną formułą. Określamy:

$$[[E]] = \begin{cases} 1 & \text{jeżeli formuła } E \text{ jest prawdziwa} \\ 0 & \text{jeżeli formuła } E \text{ jest fałszywa} \end{cases}$$

### 2.2.2 AdaBoost - ogólna idea

W pierwszym kroku algorytm AdaBoost inicjalizuje wagi dla każdej obserwacji, nadając im tę samą wartość  $\frac{1}{m}$ . Dalej następuje iteracyjne:

1. normalizowanie wag;
2. wyznaczanie klasyfikatora;

3. wyznaczanie błędu klasyfikatora;
4. wyznaczanie nowych wag na podstawie błędu klasyfikatora.

Proces iteracji kończy się, gdy błąd klasyfikatora przekroczy wartość  $\frac{1}{2}$  lub, gdy numer iteracji równy jest danej stałej  $L$  (czyli ustalonej maksymalnej liczbie iteracji). Ostateczny klasyfikator wyznaczamy na podstawie metody zwanej **głosowaniem większościowym**.

### 2.2.3 AdaBoost - algorytm

**Dane wejściowe:**

$S = \{(x_i, y_i), i = 1, 2, \dots, m\}$ ,  $y_i \in Y = \{1, \dots, K\}$  - zbiór uczący;

**LEARN** - funkcja tworząca klasyfikator;

$L$  - maksymalna liczba iteracji;

**AdaBoost(S, LEARN, L):**

1. **initialize for all**  $i : w_1(i) := \frac{1}{m}$
2. **for**  $l = 1$  **to**  $L$  **do**
3.     **for all**  $i : p_l(i) := w_l(i) / \sum_i w_l(i)$
4.      $h_l := \text{LEARN}(S, p_l)$
5.      $\varepsilon_l = \sum_i p_l(i) [[h_l(x_i) \neq y_i]]$
6.     **if**  $\varepsilon_l > \frac{1}{2}$  **then**
7.          $L := l - 1$
8.     **goto** 12
9.      $\beta_l := \varepsilon_l / (1 - \varepsilon_l)$
10.    **for all**  $i : w_{l+1}(i) := w_l(i) \beta_l^{1 - [[h_l(x_i) = y_i]]}$
11. **end for**
12. **Wyjście:**  $h_f(x) = \arg \max_{y \in Y} \sum_{l=1}^L (\log \frac{1}{\beta_l}) [[h_l(x) = y]]$

Na następnej stronie znajduje się szczegółowy opis algorytmu.

**Opis algorytmu krok po kroku:**

1. inicjalizacja wag;
2. pętla główna (iteracja: **3,4,5,6,7,8,9,10**);
3. normalizacja wag;
4. utworzenie klasyfikatora z uwzględnieniem znormalizowanych wag;
5. wyznaczenie błędu klasyfikatora;
6. jeżeli błąd przekroczył dopuszczalną wartość  $\frac{1}{2}$ , to **(7,8)** ...
7. wyznaczamy numer ostatniego klasyfikatora z błędem nie większym od  $\frac{1}{2}$ ;
8. kończymy iterację, przechodząc do reguły głosowania większościowego **(12)**;
9. wyznaczamy współczynnik  $\beta_l$ ;
10. wyznaczamy nowe wagi w oparciu o współczynnik  $\beta_l$  i klasyfikator  $h_l$ ;
11. koniec struktury pętli głównej;
12. Wyznaczamy ostateczny klasyfikator stosując metodę głosowania większościowego.

**2.3 Analiza błędów (training error)**

Główną zaletą algorytmu AdaBoost jest zdolność do redukowania błędu uczenia. *Schapire* i *Singer* pokazali, że błąd uczenia ostatecznej reguły dyskryminacyjnej  $h_f(x)$ , otrzymanej jako kombinacja klasyfikatorów  $h_l$  podczas głosowania większościowego, jest ograniczony. Jeżeli każdy klasyfikator  $h_l$  jest nieznacznie lepszy niż losowy, wtedy błąd uczenia maleje wykładniczo. To powoduje, że technika boostingu potrafi zmienić słaby algorytm uczący w niezwykle efektywny. *Freund* i *Schapire* pokazali natomiast, że uogólniony błąd końcowej reguły  $h_f(x)$  ma pewne ograniczenie górne, które zależy od:

1. błędu uczenia;
2. rozmiaru próbki uczącej;
3. wymiaru **VC** przestrzeni hipotez  $h_l$ ;
4. liczby iteracji w boostingu.

Ograniczenie to ma następującą postać:

$$Pr[h_f(x) \neq y] + \tilde{O}\left(\sqrt{\frac{Ld}{m}}\right)$$

gdzie  $Pr[\cdot]$  - oznacza prawdopodobieństwo empiryczne dla zbioru uczącego.

Alternatywną analizę błędu zaproponował *Shapire*. Posłużył się on pojęciem **margin** (margines, swoboda), które zdefiniujemy dla przypadku, gdy zbiór klas jest dwuelementowy.

**Definicja 2.2 (margin)** Dla obserwacji  $x$  z klasy  $y \in \{-1, 1\}$  definiujemy:

$$\text{margin}(x, y) = \frac{y \sum_l \alpha_l h_l(x)}{\sum_l \alpha_l}$$

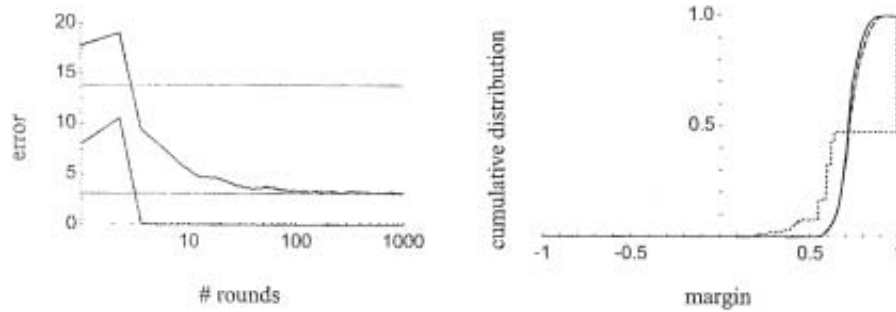
gdzie:

$$\alpha_l = \frac{1}{2} \log \left( \frac{1}{\beta_l} \right)$$

Ograniczenie przedstawione przez *Shapire*'a dla dwóch klas  $\{-1, 1\}$  ma postać:

$$Pr[\text{margin}(x, y) \leq \theta] + \tilde{O} \left( \sqrt{\frac{d}{m\theta^2}} \right) \quad \forall \theta > 0$$

Ograniczenie to nie zależy od liczby iteracji  $L$ . Udowodniono również, że boosting szczególnie koncentruje się na obiektach, które mają mały margines (margin).

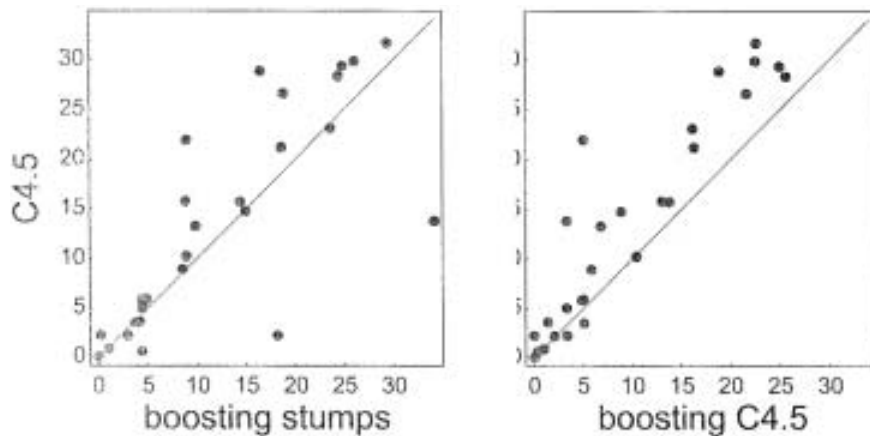


**Rysunek 1** - krzywe błędów i rozkład marginesów dla boostingu C4.5

Wykres lewy przedstawia krzywe błędów klasyfikacji dla: próby testowej (krzywa „wyższa”) i próby uczącej (krzywa „niższa”) w zależności od liczby iteracji boostingu. Linie poziome (w kolejności od „wyższej” do „niższej”) wskazują: błąd klasyfikatora podstawowego C4.5 i błąd klasyfikatora po wzmocnieniu (z zastosowaniem boostingu). Łatwo jest oczywiście zauważyć znaczną poprawę klasyfikacji.

Wykres prawy przedstawia dystrybucję empiryczną rozkładu marginesów dla zbioru uczącego w zależności od liczby iteracji boostingu:

- 5 iteracji - krzywa „kropkowana”,
- 100 iteracji - krzywa przerywana,
- 1000 iteracji - krzywa ciągła.

**Rysunek 2** - błąd C4.5 vs błąd AdaBoost dla boostingu C4.5

Rysunek 2 przedstawia porównanie błędów klasyfikacji (przy 27-elementowym zbiorze) dla:

- C4.5 vs boosting stumps (boosting stumps ???),
- C4.5 vs boosting C4.5.

Patrząc na porównanie C4.5 z boostingiem C4.5 można stwierdzić, że większość punktów znajduje się powyżej prostej identycznościowej. Wskazuje to na dużą skuteczność boostingu (błąd boostingu C4.5 jest z reguły znacznie mniejszy niż błąd C4.5).

## 2.4 Modyfikacje algorytmu AdaBoost

Istnieje kilka modyfikacji algorytmu AdaBoost. Modyfikacje te stanowią metody bardziej wyszukane, generalnie polegające na **sprowadzeniu problemu z wieloma klasami do większego problemu binarnego**. Wymienia się:

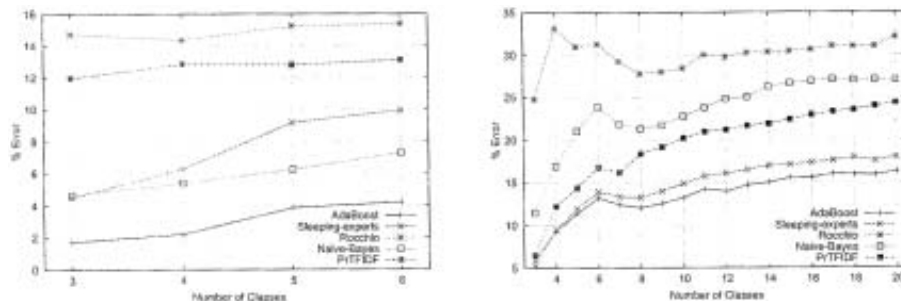
- **AdaBoost.MH** - zaproponowany przez *Schapire'a* i *Singer'a*,
- **AdaBoost.MR** - również *Schapire* i *Singer*,
- **AdaBoost.M2** - zaproponowany przez *Freund'a* i *Schapire'a* (algorytm ten jest szczególnym przypadkiem AdaBoost.MR).
- **AdaBoost.OC** - Shapire.

## 2.5 Zastosowania - przykłady

Boosting dobrze stosuje się przy rozpoznawaniu tekstu (*Shapire, Singer*). Dzięki algorytmowi AdaBoost osiąga się znacznie lepsze wyniki od innych popularnych metod (szczególnie przy identyfikacji outlier'ów):

- NaiveBayes,
- Probabilistic TF-IDF,
- Rocchio,
- Sleeping experts.

**Rysunek 3** - Porównanie AdaBoost z innymi metodami rozpoznawani tekstu.



**Rysunek 4** - Przykład rozpoznawania tekstu przez AdaBoost (kolejno wierszami 4, 12, 25 iteracji).



### 3 Podsumowanie

Istnieje przynajmniej kilka przyczyn skłaniających do konstrukcji rodzin klasyfikatorów, oraz do wniosku, że znalezienie pojedynczego klasyfikatora, który działałby tak dobrze jak rodzina, jest trudne (wręcz niemożliwe). Aby zrozumieć te przyczyny należy rozważyć naturę algorytmów uczących. Algorytmy te przeszukują przestrzeń możliwych reguł w poszukiwaniu reguły najbardziej trafnej. Bardzo ważny jest rozmiar tej przestrzeni, jak i fakt zawierania się w niej dobrego przybliżenia rozważanej zależności. Im większa jest przestrzeń reguł, tym większego zbioru uczącego potrzebujemy. Rodziny klasyfikatorów pozwalają (przy zachowaniu wymaganej precyzji) użyć zbiorów uczących o mniejszych rozmiarach.

### Literatura

- [1] Robert E. Shapire: *A Brief Introduction to Boosting*
- [2] Thomas G. Diettrich: *Machine Learning Research: Four Current Directions*
- [3] T. Hastie, R. Tibshirani, J. Friedman: *The Elements of Statistical Learning*

Praca ta dostępna jest w formie elektronicznej pod adresem:  
<http://atraktor.ask33.net/seminarium/boosting.pdf>

Wszelkie uwagi proszę kierować na adres:  
**i.glowacka@wp.pl**