

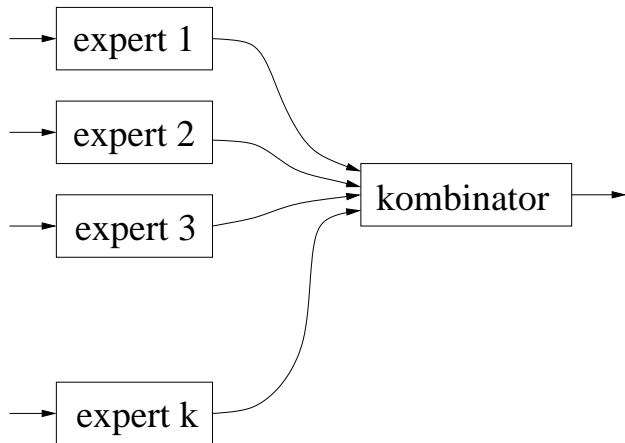
Sieci neuronowe SN1 2006/2007

Składanie klasyfikatorów

Igor T. Podolak

6 grudnia 2006

Kombinacja ekspertów



- podział błędu kwadratowego między odchylenie i wariancję

$$E[(F(x) - E[D|X = x])^2] = B^2(F(x)) + V(F(x))$$

- $B(F(x))$ to odchylenie funkcji aproksymującej — niezdolność funkcji $F(x)$ do odwzorowania funkcji $f(x) = E[D|X = x]$; to błąd *aproksymacji*

- $V(F(X))$ to wariancja funkcji aproksymującej mierzona na zbiorze uczącym — niedostatek informacji zawartej w zbiorze uczącym; to błąd *estymacji*
- odchylenie komitetu ekspertów jest takie samo jak odchylenie dla pojedynczego eksperta
- wariancja komitetu jest *mniejsza* od wariancji pojedynczego eksperta
- pojedynczy eksperci uczeni są począwszy od różnych warunków początkowych
- pojedynczy eksperci są *przeuczani*
 - poszczególni eksperci mają zredukowane odchylenie kosztem wariancji
 - komitet redukuje wariancję bez zmiany odchylenia



Bagging



Podejście przez zbudowanie szeregu klasyfikatorów \hat{f}^{*b} uczonych na zbiorach bootstrap

$$\hat{f}_{bag}(x) = \frac{1}{B} \sum_{b=1}^B \hat{f}^{*b}(x) \quad (2)$$

$\hat{f}_{bag}(x)$ zmierza do funkcji nauczanej na całym zbiorze uczącym

$$\hat{f}_{bag}(x) \longrightarrow \hat{f}(x) \text{ gdy } B \longrightarrow \infty \quad (3)$$

Pozwala na poprawienie błędu przez przeuczenie szeregu sieci

- odchylenie każdej pozostanie bez zmian
- spadnie wariancja całego układu



Primary tumor



18 atrybutów oraz klasa podająca miejsce raka:

Atrybuty:

- age: < 30 , $30 - 59$, ≥ 60
- sex: male, female
- degree-of-diffe: well, fairly, poorly
- skin: yes, no
- bone: yes, no
- bone-marrow: yes, no
- lung: yes, no
- pleura: yes, no
- peritoneum: yes, no
- liver: yes, no
- brain: yes, no
- histologic-type: epidermoid, adeno, anaplastic
- neck: yes, no
- supraclavicular: yes, no
- axillar: yes, no
- mediastinum: yes, no
- abdominal: yes, no

oraz klasa podająca miejsce

- lung, head & neck, esophagus, thyroid, stomach, duoden & sm.int, colon, rectum, anus, salivary glands, pancreas, gallbladder, liver, kidney, bladder, testis, prostate, ovary, corpus uteri, cervix uteri, vagina, breast



=== Stratified cross-validation — sieć neuronowa

- Correctly Classified Instances 130 38.3481 %
- Incorrectly Classified Instances 209 61.6519 %
- Kappa statistic 0.3078
- Mean absolute error 0.0579
- Root mean squared error 0.2042
- Relative absolute error 71.1867 %
- Root relative squared error 101.5236 %
- Total Number of Instances 339

=== Stratified cross-validation — bagging

- Correctly Classified Instances 153 45.1327 %
- Incorrectly Classified Instances 186 54.8673 %
- Kappa statistic 0.3747
- Mean absolute error 0.0573
- Root mean squared error 0.1825
- Relative absolute error 70.4899 %
- Root relative squared error 90.7133 %
- Total Number of Instances 339



Dla zadanego zbioru danych T możemy zbudować P modeli M_p . Można je połączyć wykorzystując podejście Bayesowskie z warunkowym prawdopodobieństwem modeli

$$Pr(f(x)|T) = \sum_{p=1}^P Pr(f(x)|M_p, T) Pr(M_p|T) \quad (4)$$

czy wartość oczekiwana

$$E(f(x)|T) = \sum_{p=1}^P E(f(x)|M_p, T) Pr(M_p|T) \quad (5)$$



Komitety maszyn (ang. committee) biorą zwykłą średnią poszczególnych predykcji.

Można też podejść do tego jako do sumy ważonej predykcji z wektorem wag $\hat{w} = (w_1, \dots, w_P)$

$$\hat{w} = \arg \min_w E[Y - \sum_{p=1}^P w_p \hat{f}_p(x)]^2 \quad (6)$$

rozwiązanie nie jest gorsze od żadnego z pojedynczych

$$E_T[Y - \sum_{p=1}^P w_p \hat{f}_p(x)]^2 \leq E_T[Y - \hat{f}_p(x)]^2 \quad (7)$$

Jednak praktyczne zastąpienie prostą regresją liniową może wybrać jeden model – ten o najwyższej złożoności – modele nie są równo traktowane



Primary tumor – wybór przez głosowanie



Rozwiązanie polega na zbudowaniu szeregu *różnych* klasyfikatorów (tutaj 5) i wyboru odpowiedzi przez głosowanie

- Correctly Classified Instances 137 40.413 %
- Incorrectly Classified Instances 202 59.587 %
- Kappa statistic 0.327
- Mean absolute error 0.0572
- Root mean squared error 0.189
- Relative absolute error 70.3494 %
- Root relative squared error 93.9543 %
- Total Number of Instances 339



Można wykorzystać podejście *leave-one-out* i znaleźć wektor wag na podstawie

$$\hat{w} = \arg \min_w \sum_{i=1}^N \left[y_i - \sum_{p=1}^P w_p \hat{f}_p^{-i}(x) \right]^2 \quad (8)$$

gdzie $\hat{f}_p^{-i}(x)$ jest predykcją z modelu zbudowanego na danych pomijających przykład i -ty

wtedy końcowy model

$$\hat{f}(x) = \sum_{p=1}^P w_p \hat{f}_p(x) \quad (9)$$

- bardziej złożone modele nie dostają zbyt wysokich wag
- można poprawić przez nałożenie warunków na w
 - by indywidualne wagi były nieujemne
 - by wagi sumowały się do jedności

dzięki czemu wektor wag można traktować jako wektor prawdopodobieństw poprawności poszczególnych modeli



Bumping



To właściwie metoda wyboru najlepszego z modeli

Tworzone są zbiory bootstrap T^* , a stąd odpowiednie predyktory \hat{f}^{*b}

Znajdujemy jest model, dający najmniejszy błąd predykcji uśredniony na całym zbiorze uczącym T

$$\hat{b} = \arg \min_b \sum_{i=1}^N [y_i - \hat{f}^{*b}(x_i)]^2 \quad (10)$$

Metoda staje się skuteczna gdy w zbiorze uczącym są przykłady zaburzające nauczanie i ich ominięcie może dać lepszy wynik

Jest też skuteczna gdy klasyfikatory mają problemy z wyborem początkowych parametrów co może mieć wpływ na późniejsze nauczanie – przykładem są drzewa decyzyjne

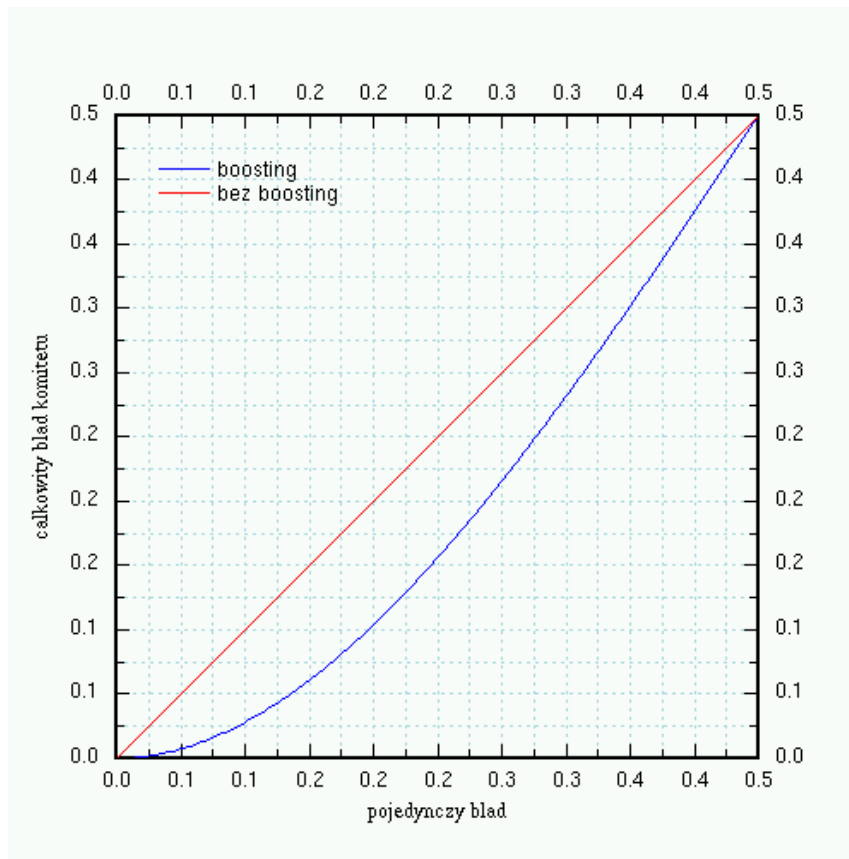


- naucz pierwszego eksperta na N_1 przykładach
- użyj pierwszego eksperta do przefiltrowania N_1 przykładów
 - rzuć monetą
 - jeśli orzeł
 - * przepuść kolejne przykłady przez pierwszego eksperta
 - * porzucaj wszystkie przykłady do momentu, gdy jeden jest źle zaklasyfikowany
 - * dodaj źle zaklasyfikowany przykład do zbioru uczącego dla drugiego eksperta
 - jeśli reszka
 - * przepuść kolejne przykłady przez pierwszego eksperta
 - * porzucaj wszystkie przykłady do momentu, gdy jeden jest dobrze zaklasyfikowany
 - * dodaj dobrze zaklasyfikowany przykład do zbioru uczącego dla drugiego eksperta
- naucz drugiego eksperta korzystając z N_2 przykładów
- po nauczaniu przefiltruj kolejne N_1 przykładów korzystając z obydwu ekspertów
 - jeśli obaj eksperci zgadzają się, to porzuć ten przykład
 - jeśli nie zgadzają się, to dodaj go do zbioru uczącego dla trzeciego eksperta
- naucz trzeciego eksperta używając N_3 przykładów

Wynik klasyfikacji uzyskiwany jest przez *głosowanie* lub *uśrednianie*

Boosting przez filtrowanie

- potrzeba $3N_1$ przykładów, jednak do obliczeń wykorzystywane jest $N_1 + N_2 + N_3 < 3N_1$ przykładów, w tym sensie algorytm jest “sprytny”
- algorytm wymaga jednak dostępności do bardzo dużej liczby przykładów

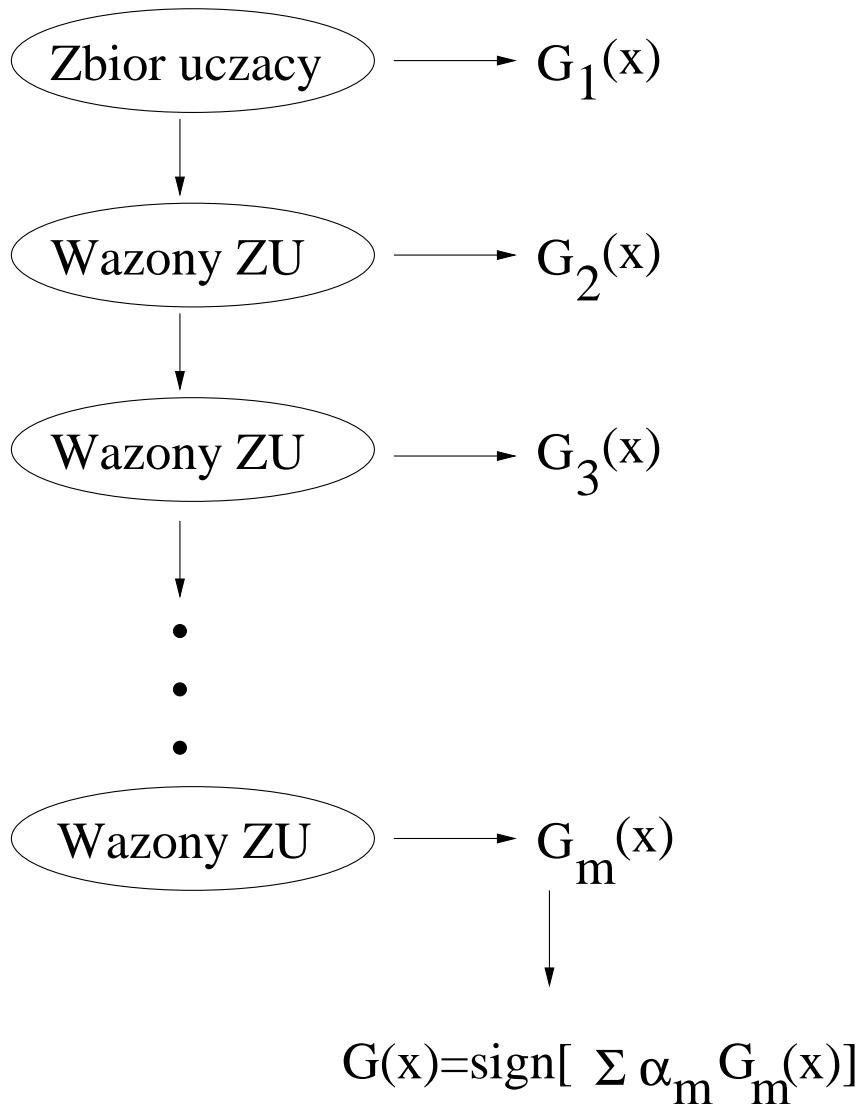


- dzięki filtrowaniom kolejni eksperci skupiają się na “trudnych” do nauczenia fragmentów rozkładu danych
- jeśli poszczególni eksperci mają błąd $\epsilon < 1/2$, to błąd komitetu jest ograniczony (Shapire, 1990) przez

$$g(\epsilon) = 3\epsilon^2 - 2\epsilon^3 \quad (11)$$

- w ten sposób ze *słabych* modeli tworzony jest jeden *silny*

AdaBoost – Adaptive Boosting



- tworzy szereg zbiorów uczących i klasyfikatory oparte na nich
- “trudniejsze” przykłady otrzymują w późniejszych zbiorach uczących większą wagę
- procedurę można zastosować nawet gdy nie jest dostępnych bardzo wiele przykładów



AdaBoost – algorytm dla problemu dwu-klasowego



- inicjalizacja wag wszystkich przykładów na $w_i = 1/N$, $i = 1, 2, \dots, N$ — z tym prawdopodobieństwem każdy przykład będzie losowany do nauczania
- dla $m = 1$ do M znajdź klasyfikator $G_m(x)$ dla aktualnego zbioru przykładów, gdzie wagi odpowiadają rozkładowi prawdopodobieństwa z jakim przykłady będą wybierane do nauczania



AdaBoost – algorytm dla problemu dwu-klasowego



- oblicz ważony błąd aktualnego klasyfikatora i zbuduj nowy zbiór uczący

$$err_m = \sum_{i=1}^N w_i I(y_i \neq G_m(x_i)) / \sum_{i=1}^N w_i \quad (12)$$

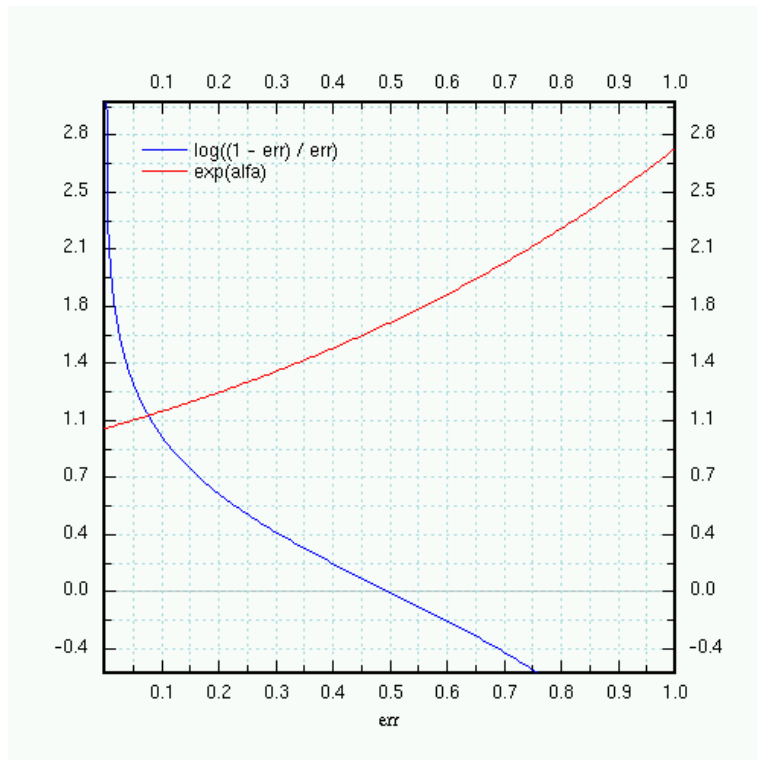
$$\alpha_m = \log((1 - err_m) / err_m) \quad (13)$$

$$w_i = w_i \exp[\alpha_m I(y_i \neq G_m(x_i))] \quad (14)$$

- wygeneruj końcowy klasyfikator

$$G(x) = \text{sign} \left[\sum_{i=1}^N \alpha_m G_m(x) \right] \quad (15)$$

- α_m obrazuje skuteczność klasyfikatora względem klasyfikatora przypadkowego – dla klasyfikatora gorszego od losowego wartość spada poniżej 0





==== Stratified cross-validation ====

- Correctly Classified Instances 146 43.0678 %
- Incorrectly Classified Instances 193 56.9322 %
- Kappa statistic 0.3545
- Mean absolute error 0.0588
- Root mean squared error 0.1913
- Relative absolute error 72.3786 %
- Root relative squared error 95.1055 %
- Total Number of Instances 339



Krokowe modelowanie addytywne



AdaBoost jest typowym krokowym modelowaniem addytywnym: w każdym kroku iteracji wyszukiwana jest optymalna funkcja podstawowa $b(x_i; \gamma_m)$ i dodawana do aktualnego rozwinięcia $f_{m-1}(x)$ **bez** zmiany poprzednich elementów

- Inicjalizacja $f_0(x) = 0$
- dla $m = 1$ do M

$$(\beta_m, \gamma_m) = \arg \min_{\beta, \gamma} \sum_{i=1}^N L(y_i, f_{m-1}(x_i) + \beta b(x_i; \gamma)) \quad (16)$$

$$f_m(x) = f_{m-1}(x) + \beta_m b(x; \gamma_m) \quad (17)$$



Dla kwadratowej funkcji kosztu mamy

$$L(y, f(x)) = (y - f(x))^2 \quad (18)$$

$$L(y_i, f_{m-1}(x_i) + \beta b(x_i; \gamma)) = (y_i - f_{m-1}(x_i) - \beta b(x_i; \gamma))^2 \quad (19)$$

$$= (r_{im} - \beta b(x_i; \gamma))^2 \quad (20)$$

Czynnik $r_{im} = y_i - f_{m-1}(x_i)$ to błąd aktualnego modelu na przykładzie i . Składnik $\beta b(x; \gamma)$ stara się go zminimalizować.



Definiujemy *eksponencjalną funkcję kosztu*

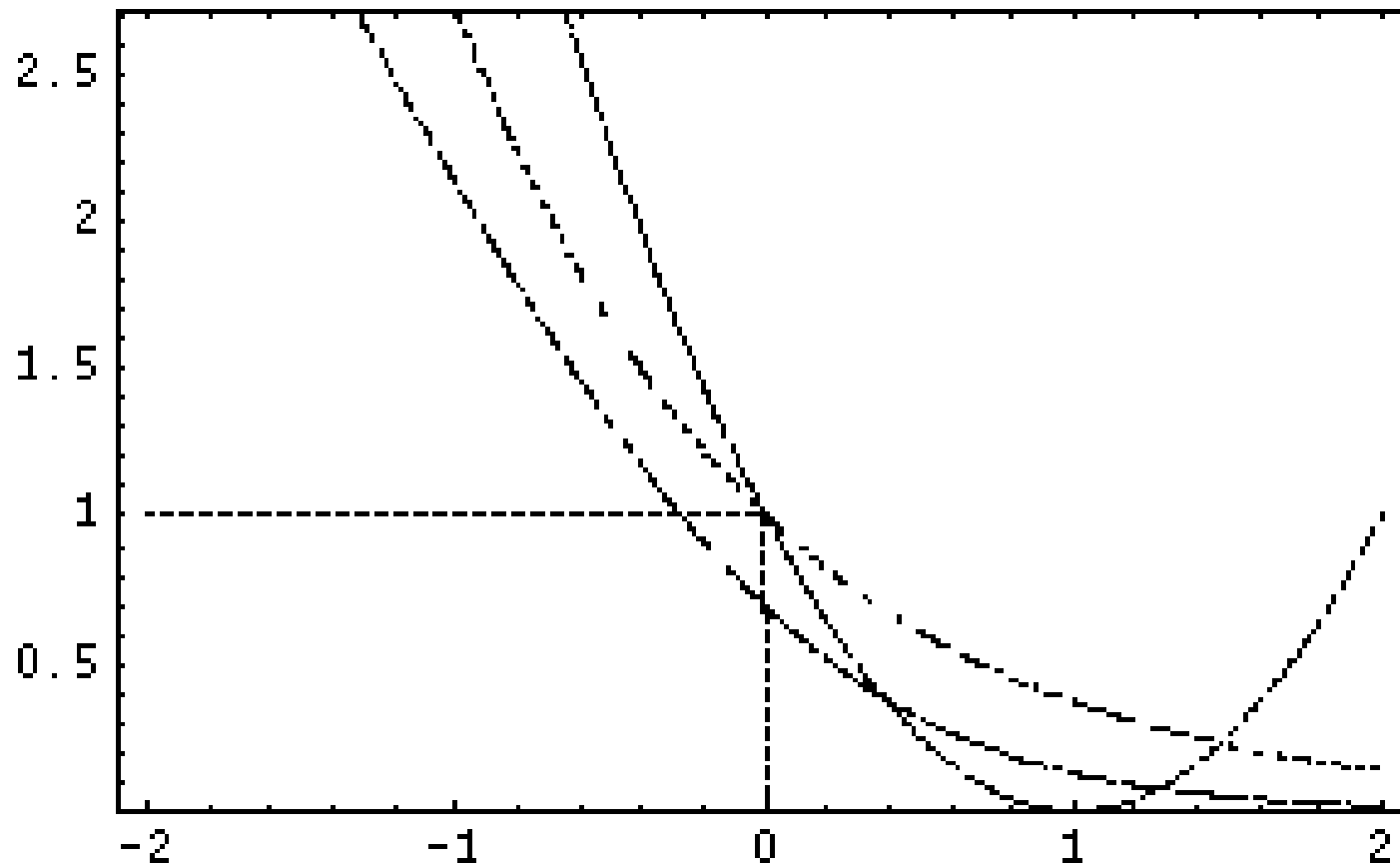
$$L(y, f(x)) = \exp(-yf(x)) \quad (21)$$

ponieważ

- ułatwia to obliczenia zmian wag w modelu AdaBoost
- funkcja eksponencjalna odpowiada określaniu przez AdaBoost log-odd jednej z klas: funkcja minimalizująca f^* określa

$$Pr(Y = 1|x) = \frac{1}{1 + e^{-2f^*(x)}} \quad (22)$$

Eksponencjalna funkcja kosztu



Rysunek 1: Funkcje kosztu dla problemu klasyfikacji o odpowiedzi $y \in \{-1, +1\}$: błędnej klasyfikacji (nieciągła), kwadratowa (z minimum w 1), eksponencjalna (malejąca przechodząca przez punkt $(0, 1)$), logistyczna (malejąca przechodząca poniżej punktu $(0, 1)$)

- dla klasyfikacji wartość $yf(x)$ pełni tą samą rolę co $y - f(x)$ w przypadku regresji
- jeśli przykład jest źle klasyfikowany to $yf(x)$, jeśli to dodatnia
- prawidłowa funkcja kosztu powinna karać ujemne wartości
- w większym stopniu karze za błędy niż “nagradza” za poprawne postępowanie (podobnie jak logistyczna $\log(1 + e^{-f(x)})$)
- nieliniowa funkcja niepoprawnej klasyfikacji nie karze za poprawne klasyfikacje, a za wszystkie błędne karze jednakowo
- wzrost kary jest dla funkcji eksponencjalnej jest eksponencjalny, podczas gdy dla logistycznej staje się prawie liniowy dla dużych wartości błędu

- funkcja eksponencjalna w znacznie większym stopniu niż inne karze za błędy
- \hat{f} wykorzystujący funkcję eksponencjalną będzie minimalizować wartość oczekiwaną błędu $\exp(-yf(x))$ (dla klasyfikatora dającego odpowiedzi $\{-1, +1\}$)

$$\hat{f}(x) = \arg \min_{f(x)} E[\exp(Y f(x))] = \frac{1}{2} \frac{Pr(Y = 1|x)}{Pr(Y = -1|x)} \quad (23)$$

- w przypadku funkcji kwadratowej

$$\hat{f}(x) = \arg \min_{f(x)} E[(Y - f(x))^2] = E[Y|x] = 2Pr(Y = 1|x) - 1 \quad (24)$$

- dla problemów klasyfikacji funkcja eksponencjalna jest często lepszym wyborem



Korzystając z klasyfikatorów $G_m(x) \in \{-1, +1\}$ potrzeba rozwiązać

$$(\beta_m, G_m) = \arg \min_{\beta, G} \sum_{i=1}^N \exp[y_i(f_{m-1}(x_i) + \beta G(x))] \quad (25)$$

by znaleźć współczynniki odpowiadające za minimalizację złożonego predyktora

$$(\beta_m, G_m) = \arg \min_{\beta, G} \sum_{i=1}^N w_i^{(m)} \exp[-\beta y_i G(x_i)] \quad (26)$$

$$w_i^{(m)} = \exp(-y_i f_{m-1}(x_i)) \quad (27)$$

w_i jest *niezależne* od β i G – można na nie spojrzeć jak na wagę dodaną do każdej obserwacji



Dla dowolnego $\beta > 0$ rozwiązanie ma postać

$$G_m = \arg \min_G \sum_{i=1}^N w_i^{(m)} I(y_i \neq G(x_i)) \quad (28)$$

G_m jest klasyfikatorem minimalizującym ważony błąd, stąd (26) można zapisać jako

$$(\beta_m, G_m) = \arg \min_{\beta, G} \sum_{i=1}^N w_i^{(m)} \exp[-\beta y_i(x_i)] \quad (29)$$

$$= e^{-\beta} \sum_{y_i=G(x_i)} w_i^{(m)} + e^{\beta} \sum_{y_i \neq G(x_i)} w_i^{(m)} \quad (30)$$

co jest innym sposobem wyrażenia warunku

$$(\beta_m, G_m) = \arg \min_{\beta, G} \sum_{i=1}^N w_i^{(m)} \exp[-\beta y_i(x_i)] \quad (31)$$

$$(32)$$



$$(\beta_m, G_m) = \arg \min_{\beta, G} \sum_{i=1}^N w_i^{(m)} \exp[-\beta y_i(x_i)] \quad (33)$$

$$= e^{-\beta} \sum_{y_i=G(x_i)} w_i^{(m)} + e^{\beta} \sum_{y_i \neq G(x_i)} w_i^{(m)} \quad (34)$$

$$= e^{\beta} \sum_{y_i \neq G(x_i)} w_i^{(m)} - e^{-\beta} \sum_{y_i \neq G(x_i)} w_i^{(m)} \quad (35)$$

$$+ e^{-\beta} \sum_{y_i \neq G(x_i)} w_i^{(m)} + e^{-\beta} \sum_{y_i=G(x_i)} w_i^{(m)} \quad (36)$$

$$= (e^{\beta} - e^{-\beta}) \sum_{y_i \neq G(x_i)} w_i^{(m)} + e^{-\beta} \sum_{i=1}^N w_i^{(m)} \quad (37)$$

$$= (e^{\beta} - e^{-\beta}) \sum_{i=1}^N w_i^{(m)} I(y_i \neq G(x_i)) + e^{-\beta} \sum_{i=1}^N w_i^{(m)} \quad (38)$$



Z tej postaci G_m można znaleźć β minimalizujące błąd

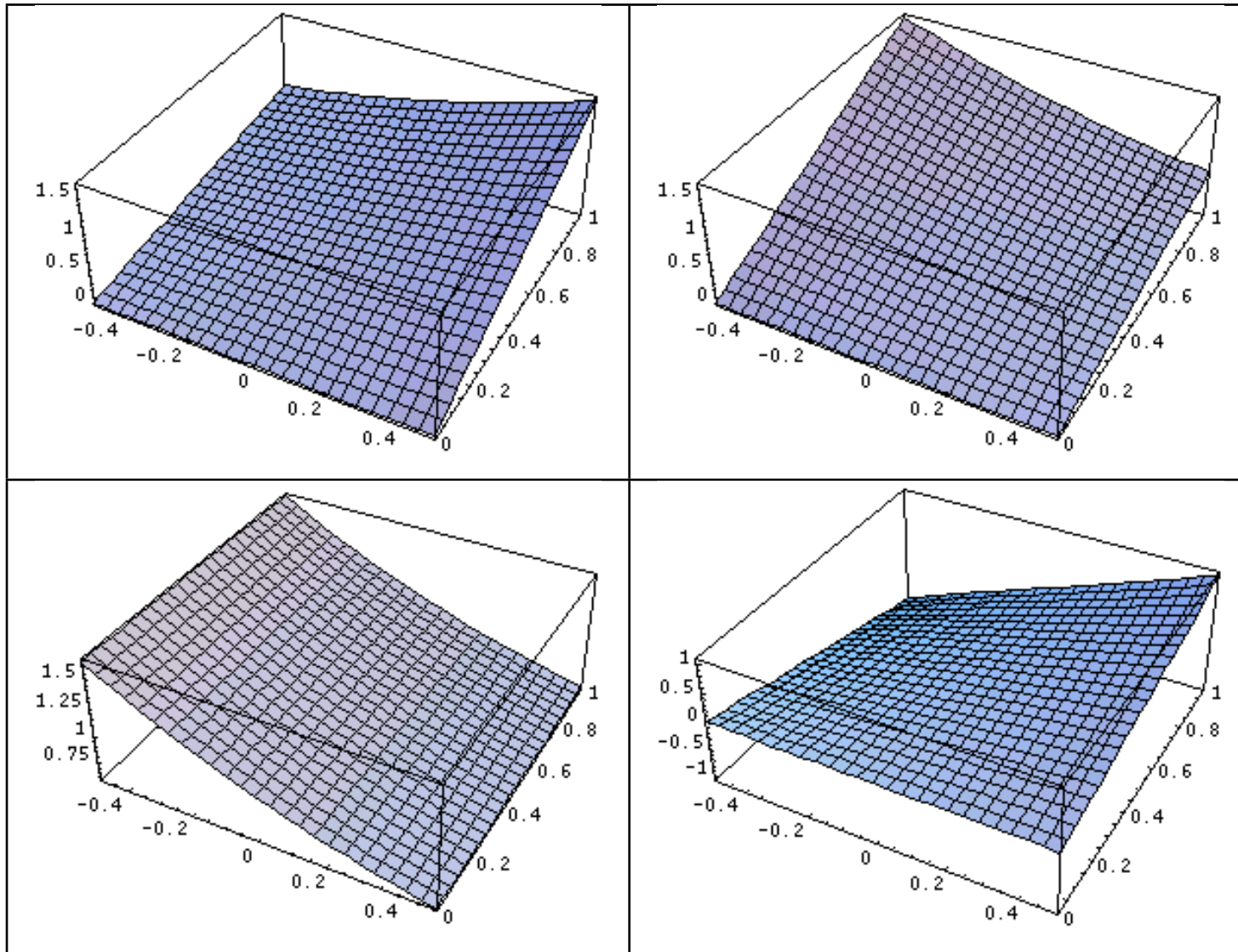
$$e^{\beta} \sum_{i=1}^N w_i^{(m)} I(y_i \neq G(x_i)) - e^{-\beta} \sum_{i=1}^N w_i^{(m)} I(y_i \neq G(x_i)) \quad (39)$$

$$+ e^{-\beta} \sum_{i=1}^N w_i^{(m)} \quad (40)$$

która jest dodatnio określona dla błędu z przedziału $[0, 1]$, a drugą pochodną po β ma dodatnią (druga pochodna jest równa samej funkcji)

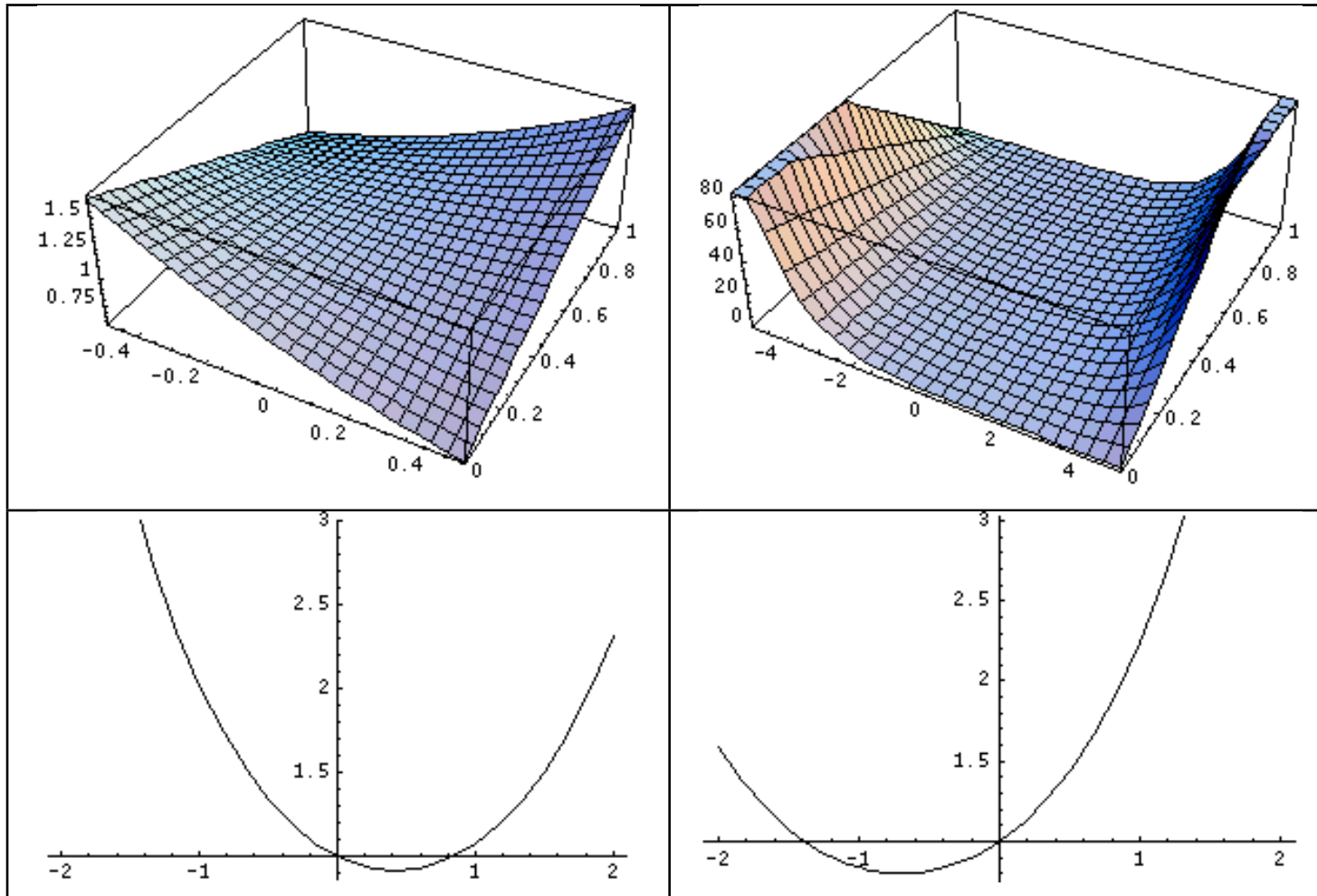


AdaBoost jako krokowe modelowanie addytywne





AdaBoost jako krokowe modelowanie addytywne





$$e^{\beta} \sum_{i=1}^N w_i^{(m)} I(y_i \neq G(x_i)) - e^{-\beta} \sum_{i=1}^N w_i^{(m)} I(y_i \neq G(x_i)) \quad (41)$$

$$+ e^{-\beta} \sum_{i=1}^N w_i^{(m)} \quad (42)$$

$$\frac{\partial G}{\partial \beta} = e^{\beta} \sum_{i=1}^N w_i^{(m)} I(y_i \neq G(x_i)) + e^{-\beta} \sum_{i=1}^N w_i^{(m)} I(y_i \neq G(x_i)) \quad (43)$$

$$- e^{-\beta} \sum_{i=1}^N w_i^{(m)} = 0 \quad (44)$$

$$\left(\frac{e^{\beta}}{e^{-\beta}}\right) \frac{\sum_{i=1}^N w_i^{(m)} I(y_i \neq G(x_i))}{\sum_{i=1}^N w_i^{(m)}} + \left(\frac{e^{-\beta}}{e^{-\beta}}\right) \frac{\sum_{i=1}^N w_i^{(m)} I(y_i \neq G(x_i))}{\sum_{i=1}^N w_i^{(m)}} - 1 = 0 \quad (45)$$

$$e^{2\beta} err_m + err_m - 1 = 0 \quad (46)$$

$$\beta_m = \frac{1}{2} \log \frac{1 - err_m}{err_m} \quad (47)$$

$$err_m = \frac{\sum_{i=1}^N w_i^{(m)} I(y_i \neq G(x_i))}{\sum_{i=1}^N w_i^{(m)}} \quad (48)$$



Zgodnie z modelem addytywnym modyfikowane są modele, a wobec tego wagi przykładów do kolejnej poprawy są uaktualniane

$$f_m(x) = f_{m-1}(x) + \beta_m G_m(x) \quad (49)$$

$$w_i^{(m+1)} = w_i^{(m)} \exp(-\beta y_i G_m(x)) \quad (50)$$

$$-y_i G_m(x) = 2I(y_i \neq G(x_i)) - 1 \quad (51)$$

$$w_i^{(m+1)} = w_i^{(m)} \exp(-2\beta_m I(y_i \neq G(x_i))) \exp(-\beta_m) \quad (52)$$

$$w_i^{(m+1)} = w_i^{(m)} \exp(-\alpha_m I(y_i \neq G(x_i))) \exp(-\beta_m) \quad (53)$$

dla $\alpha_m = 2\beta_m$, co jest zgodne z algorytmem AdaBoost (czynniki $\exp(-\beta_m)$ jest stały).



- AdaBoost demo <http://www.cs.huji.ac.il/~yoavf/adaboost/index.html> (klasyfikacja punktów na płaszczyźnie)