Podstawy Sztucznej Inteligencji (PSZT)

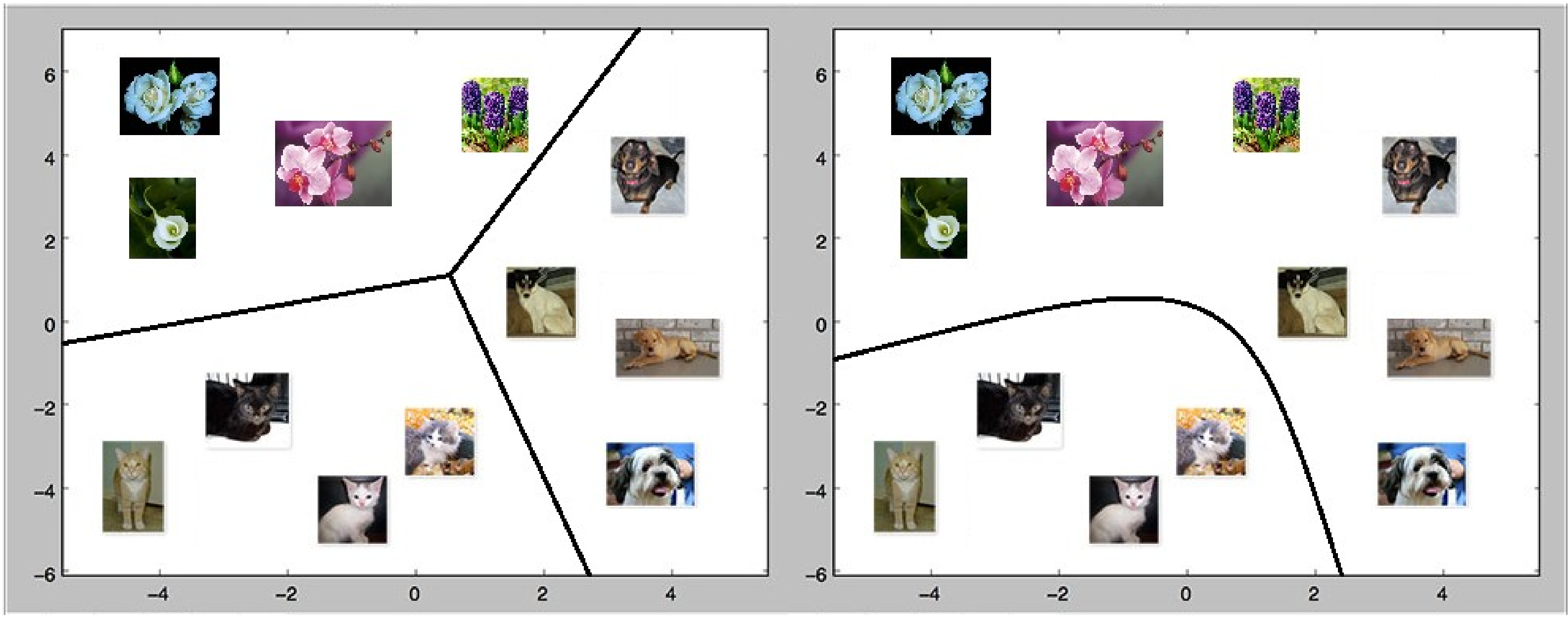
Paweł Wawrzyński

Uczenie maszynowe
Klasyfikacja i klasyfikatory

Plan na dziś

- Maszyny wektorów nośnych
- Drzewa decyzyjne i lasy losowe
- Gradient boosting

Maszyny wektorów nośnych
Support Vector Machines – SVM



- Klasyfikator
- Na podstawie danych buduje funkcję

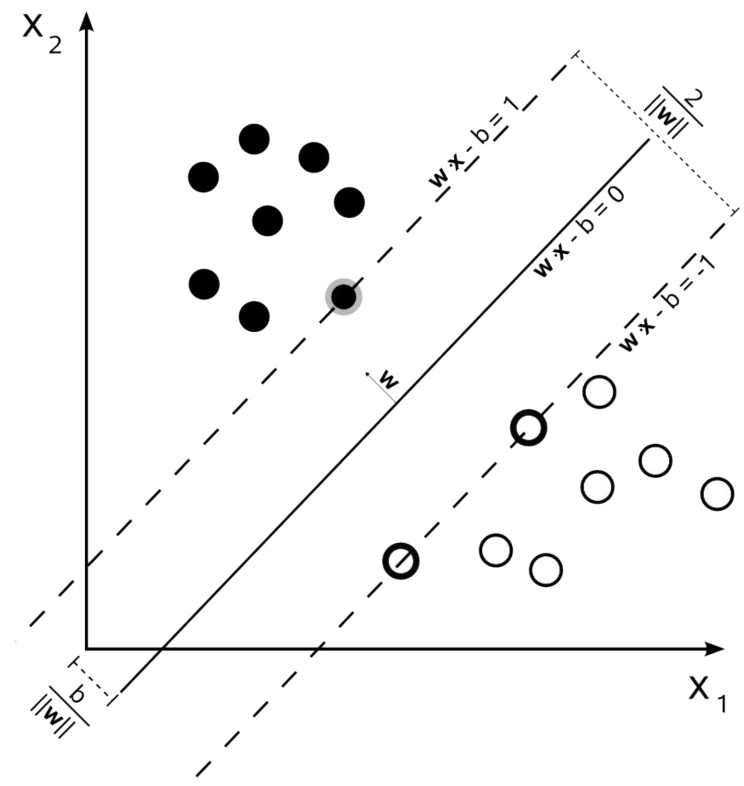
$$f(x) > 0 \rightarrow x \in \text{Klasa}$$
$$f(x) \leq 0 \rightarrow x \notin \text{Klasa}$$

SVM – przypadek liniowo separowalny

x_i - i-ty obraz

$y_i = 1$ jeśli $x_i \in \text{Klasa}$

$y_i = -1$ jeśli $x_i \notin \text{Klasa}$



Funkcja rozgraniczająca

$f(x) = w^T x - b$

$(w, b) = \arg\min_{w,b} \|w\|^2$

przy ograniczeniach

$w^T x_i - b \geq 1$ dla $x_i \in \text{Klasa}$

$w^T x_i - b \leq -1$ dla $x_i \notin \text{Klasa}$

inaczej, przy ograniczeniach

$(w^T x_i - b)\, y_i \geq 1$

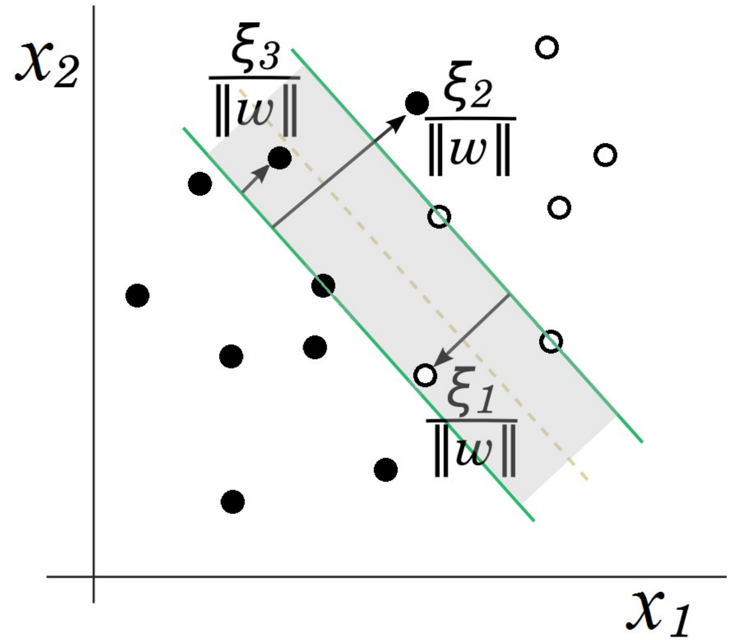
SVM – przypadek nieseparowalny liniowo

$$f(x) = w^T x - b$$

$$(w, b) = \underset{w,b}{\arg\min} \sum_i \xi_i + \lambda \|w\|^2$$

$$\lambda > 0$$

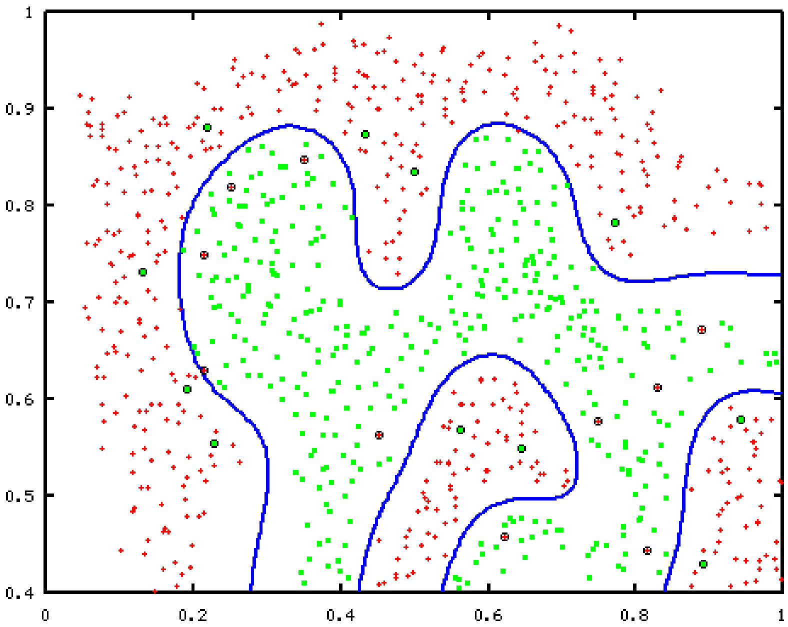
przy ograniczeniach
dla każdego i:
$\xi_i \geq 0$
$(w^T x_i - b) y_i \geq 1 - \xi_i$



Wniosek z powyższego
$\xi_i = \max\{1 - f(x_i) y_i, 0\}$

Twierdzenie o reprezentacji
$w = \sum_i \alpha_i y_i x_i$
$\alpha_i \neq 0$ tylko dla $i \in SVs$

SVM – postać nieliniowa



- Zasada taka sama, ale nowa przestrzeń

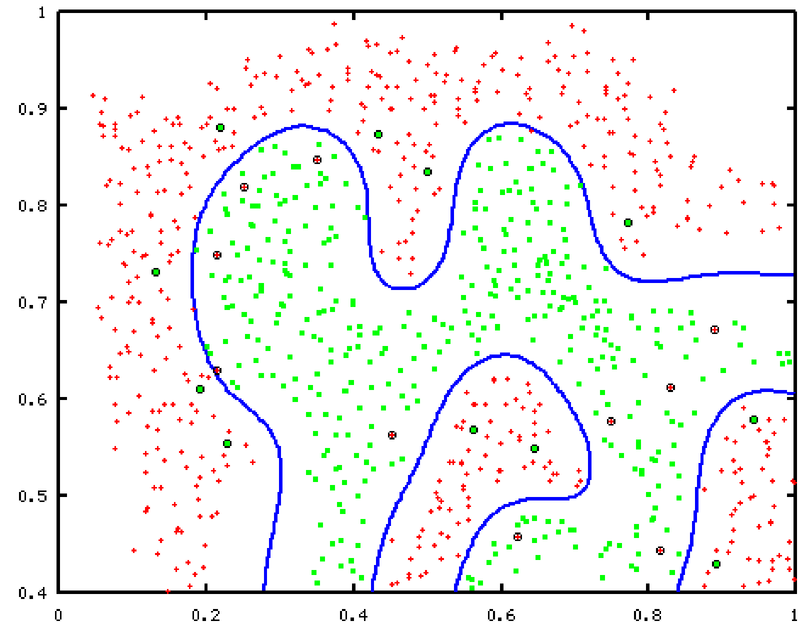
$$z = \phi(x)$$

$$w = \sum_i \alpha_i y_i \phi(x_i)$$

$$f(x) = w^T \phi(x) - b$$

$$(\alpha_{1..N}, b) = \arg\min_{\alpha_{1..N}, b} \sum_i \max\{1 - f(x_i) y_i, 0\} + \lambda \|w\|^2$$

SVM – postać nieliniowa



$$f(x) = w^T \phi(x) - b$$
$$w = \sum_i \alpha_i y_i \phi(x_i)$$
$$f(x) = \sum_i \alpha_i y_i \phi(x_i)^T \phi(x) - b$$

Jądra (*kernels*) SVM

$$\phi(x)^T \phi(y) = k(x, y)$$

liniowe: $k(x, y) = x^T y$

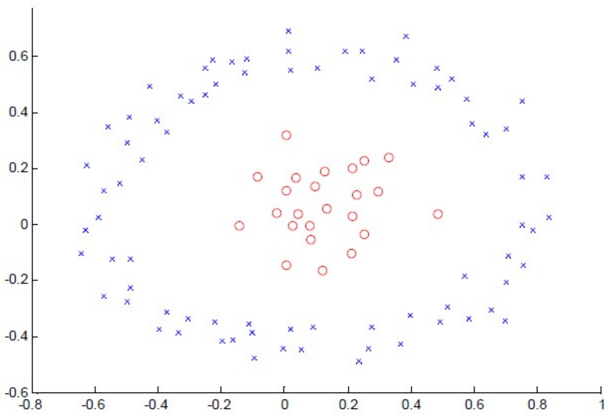
wielomianowe: $k(x, y) = (1 + x^T y)^d, d > 0$

gaussowskie (RBF): $k(x, y) = \exp\left(-\|x - y\|^2 / 2\sigma^2\right)$

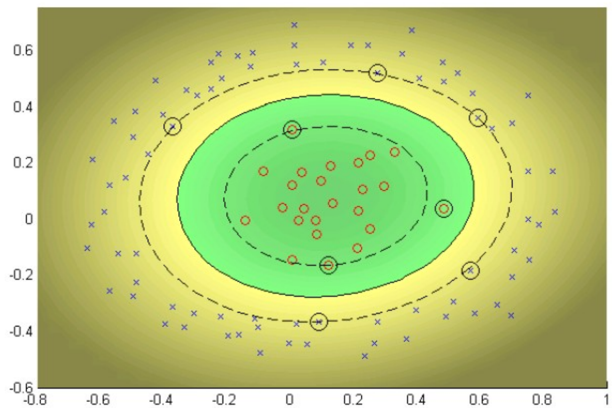
SVM – jądro RBF

$$(\alpha_{1..N}, b) = \underset{\alpha_{1..N}, b}{\arg\min} \sum_i max\{1 - f(x_i) y_i, 0\} + \lambda \|w\|^2$$

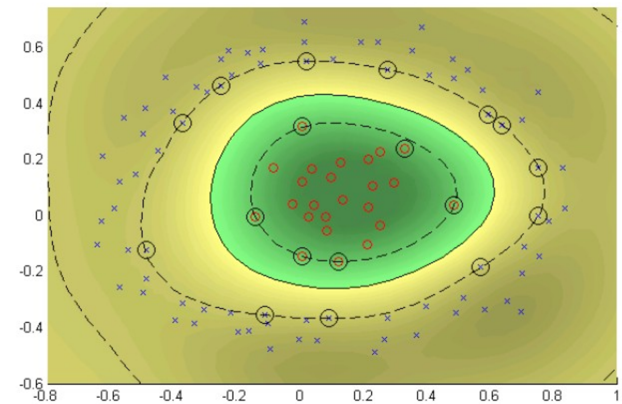
$$f(x) = \sum_i \alpha_i y_i \exp(-\|x_i - x\|^2 / 2\sigma^2)$$



$\sigma = 1, \lambda = 10$

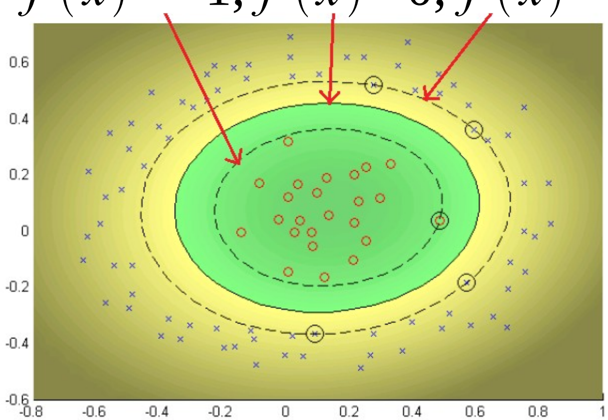


$\sigma = 0.25, \lambda \approx \infty$

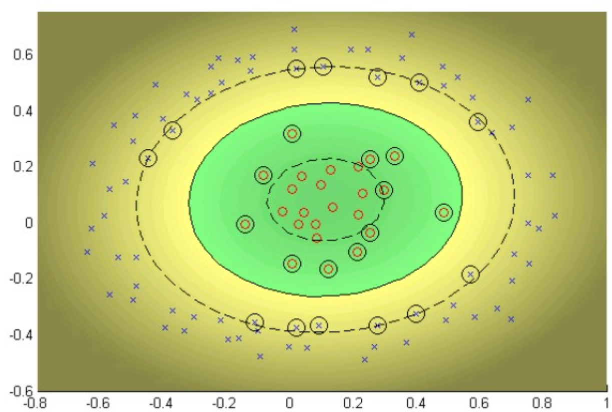


$\sigma = 1, \lambda = 1$

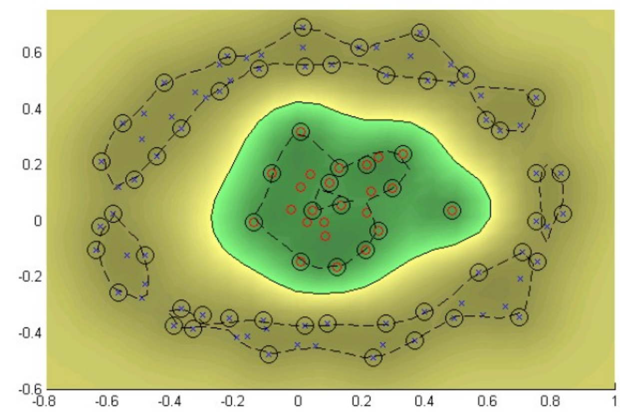
$f(x) = -1, f(x) = 0, f(x) = 1$



$\sigma = 1, \lambda = 100$

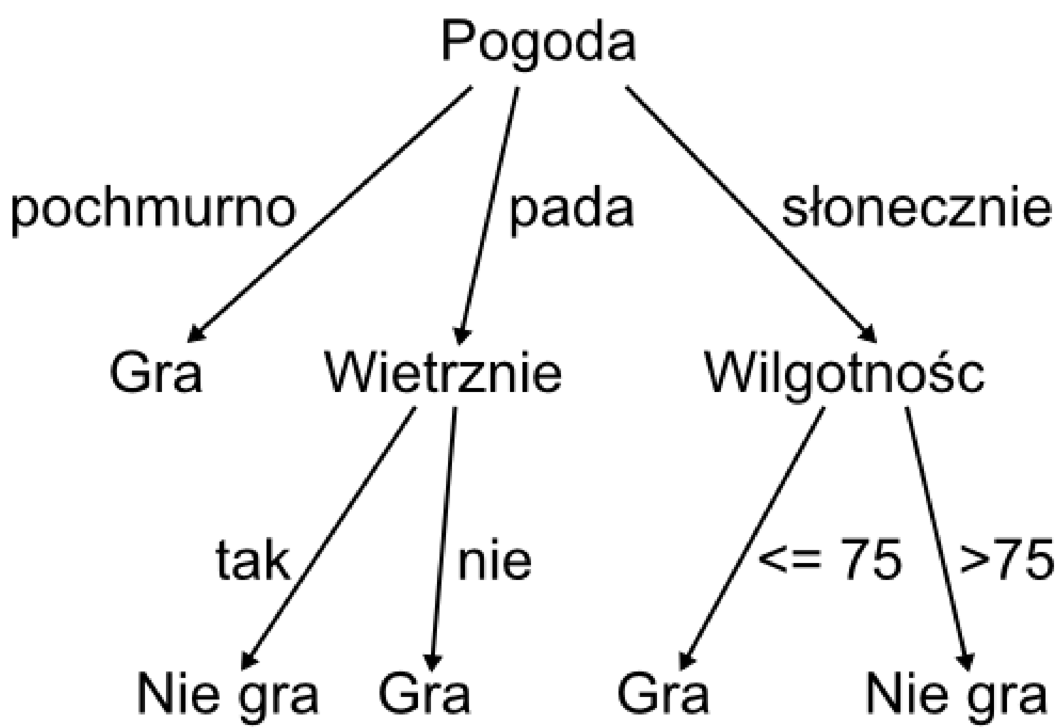


$\sigma = 0.1, \lambda \approx \infty$



Indukcja drzew decyzyjnych

- Baza danych przypadków z ich klasyfikacją
- Na jej podstawie
  → mechanizm decyzyjny w formie drzewa
- Czy Tiger Woods
  gra w golfa?



Algorytm ID3

funkcja **ID3**(C : zbiór klas,
R : zbiór atrybutów poza klasą,
S : zbiór obiektów)

jeśli $S = \emptyset$: zwróć błąd
jeśli wszystkie obiekty w S są tej samej klasy:
    zwróć węzeł zawierający tylko tę klasę
jeśli $R = \emptyset$:
    zwróć węzeł zawierający klasę najczęstszą w S
$D =$ atrybut maksymalizujący $InfGain(D, S)$
$d_j = j-$ta wartość tego atrybutu, $j = 1,2,\ldots$
$S_j = \{o \in S \mid D(o) = d_j\}$
zwróć drzewo z korzeniem oznaczonym przez D,
krawędziami d_j, $j = 1,2,\ldots$, prowadzącymi do drzew
$ID3(R - \{D\}, C, S_1), ID3(R - \{D\} S_2), \ldots$

InfGain

- Entropia zbioru, f_i - częstość *i*-tej klasy

$$I(S) = -\sum_i f_i \ln f_i$$

- Entropia zbioru podzielonego na podzbiory

$$Inf(D, S) = \sum_j \frac{|S_j|}{|S|} I(S_j)$$

- Zdobycz informacyjna

$$InfGain(D, S) = I(S) - Inf(D, S)$$

Algorytm C4.5

funkcja **C4.5**(C : zbiór klas,
R : zbiór atrybutów poza klasą,
S : zbiór obiektów)

$T = \boldsymbol{ID3}$
Dla każdego liścia T:
    Dla każdego węzła w na drodze liść-korzeń:
        $e_0 =$ oszacowanie błędu testowego w poddrzewie w
        $e_1 =$ oszacowanie błędu testowego,
            gdyby w zwracał najczęstszą w nim klasę
        jeśli $e_0 \geq e_1$:
            zastąp poddrzewo liściem zwracającym
            najczęstszą w nim klasę

Szacowanie błędu testowego

$e_S -$ błąd na zbiorze treningowym

$e_T -$ szacowany błąd na zbiorze testowym

$$e_T \approx e_S + \frac{\sqrt{e_S(1-e_S)}}{|S|}$$

W stronę lasów losowych

- Wadą ID3 jest przeuczenie (*overfitting*)
- Wadą C4.5 jest nienajlepsza klasyfikacja ← zachłanny sposób wyboru atrybutów w węzłach drzewa
- Pomysł:
  - Zbudować wiele różnych drzew na bazie różnych problemów niesprzecznych z danym
  - Klasyfikacja ← dominanta klasyfikacji dokonywanych przez drzewa

Budowa lasu losowego (*random forest*)

procedura **Twórz_las**(C: zbiór klas,
R: zbiór atrybutów poza klasą,
S: zbiór obiektów)

1: dla $b = 1, \ldots, B$:
2: $S_b = B$ obiektów z S wylosowanych ze zwracaniem
3: $R_b = \lfloor \sqrt{|R|} \rfloor$ atrybutów wylosowanych bez zwracania z R
4: $f_b =$ drzewo decyzyjne zbudowane na podstawie C, R_b, S_b

funkcja **Zaklasyfikuj**(o: obiekt)
1: dla $b = 1, \ldots, B$:
2: $c_b =$ klasa obiektu o wskazana przez f_b
3: zwróć najczęstszą klasę w $\{c_1, \ldots, c_B\}$

Gradient Boosting – idea

- Zadanie aproksymacji na zbiorze skończonym $\langle x_i, y_i \rangle,\ i \in \{1, \ldots, n\}$
- Modele

$$\bar{f}_m : R^{n_x} \rightarrow R^{n_y},\ \ \gamma_m \in R,\ \ \bar{F}_m(x_i) = \sum_m \gamma_m \bar{f}_m(x_i) \approx y_i$$

- Funkcja straty

$$q_i : R^{n_y} \rightarrow R,\ \text{np. } q_i(y) = 0.5 \| y - y_i \|^2$$

- W pętli:
  - Kolejne $\bar{f}_m$ poprawia błędy dotychczasowego modelu

Gradient Boosting – algorytm

1: Inicjalizacja wartością stałą

$$F_0(x) \equiv \arg\min_{\gamma} \sum_{i=1}^{n} q_i(\gamma).$$

2: Dla $m=1$ do M:

2.1. Oblicz pseudo-rezidua:

$$r_{i,m} = -\left[\frac{\partial q_i(F_{m-1}(x_i))}{\partial F_{m-1}(x_i)}\right], \quad i=1,\ldots,n, \text{ np. } r_{i,m} = y_i - F_{m-1}(x_i)$$

2.2. Naucz $\bar{f}_m$ używając $\langle x_i, r_{i,m} \rangle, i=1,\ldots,n$ jako zbioru treningowego

2.3. Oblicz γ_m

$$\gamma_m = \arg\min_{\gamma} \sum_{i=1}^{n} q_i(F_{m-1}(x_i) + \gamma \bar{f}_m(x_i)).$$

2.4. $F_m(x) = F_{m-1}(x) + \gamma_m \bar{f}_m(x)$

3. Zwróć $F \equiv F_M$

XGBoost – biblioteka

- eXtreme Gradient Boosting
- Algorytm: Gradient Boosting
- $\bar{f}_m$ mają postać drzew
- Do ściągnięcia z github-a
- Projekt rozpoczęty przez Tianqi Chen'a z Distributed Machine Learning Community
- Często wygrywa konkursy na Kaggle.com
- „When in doubt, use xgboost”