

◆ 1. AdaBoost — Boosting

✦ Idea

Boosting to metoda łączenia wielu słabych klasyfikatorów (np. decision stumps) w silny klasyfikator zespołowy poprzez ważenie klasyfikatorów w oparciu o ich jakość.

✦ Algorytm AdaBoost

1 Inicjalizacja wag:

$$w_i^{(1)} = \frac{1}{n}, \quad i = 1, \dots, n$$

2 Iteracje $k = 1, \dots, B$:

- A. Uczenie klasyfikatora $f_k(x)$ z uwzględnieniem wag $w_i^{(k)}$.
- B. Błąd ważony:

$$\varepsilon_k = \sum_{i=1}^n w_i^{(k)} \mathbf{1}\{f_k(x_i) \neq y_i\}$$

- C. Współczynnik:

$$\beta_k = \frac{\varepsilon_k}{1 - \varepsilon_k}$$

- D. Aktualizacja wag:

$$w_i^{(k+1)} = \begin{cases} w_i^{(k)} \cdot \beta_k, & \text{jeśli } f_k(x_i) = y_i \\ w_i^{(k)}, & \text{w przeciwnym razie} \end{cases}$$

a następnie normalizacja wag:

$$w_i^{(k+1)} = \frac{w_i^{(k+1)}}{\sum_{j=1}^n w_j^{(k+1)}}$$

3 Predykcja:

- Klasyfikator zespołowy:

$$\hat{y}(x) = \arg \max_y \sum_{k=1}^B \mathbf{1}\{f_k(x) = y\} \log \left(\frac{1}{\beta_k} \right)$$

- Każdy klasyfikator głośnie proporcjonalnie do $\log(1/\beta_k)$.

✓ Im mniejszy błąd ε_k , tym większy wpływ klasyfikatora w głosowaniu.

♦ 2. Bagging — Bootstrap Aggregating

✦ Idea

Bagging redukuje wariancję poprzez uczenie wielu modeli na różnych próbkach bootstrapowych (z powtórzeniami) i uśrednianie predykcji.

✦ Algorytm

- 1 Tworzymy B bootstrapowych próbek ze zbioru treningowego.
- 2 Dla każdej próbki uczymy klasyfikator (np. drzewo decyzyjne).
- 3 Predykcja to głosowanie większościowe:

$$\hat{y}(x) = \text{majority vote}\{f_1(x), f_2(x), \dots, f_B(x)\}$$

♦ 3. Random Forest

✦ Idea

Rozszerzenie baggingu: dodatkowo w każdej nodze drzewa wybierane jest losowe podzbiór cech (np. \sqrt{p}), co zwiększa różnorodność klasyfikatorów i zmniejsza korelację między drzewami.

♦ 4. XGBoost

✦ Idea

XGBoost (Extreme Gradient Boosting) to algorytm boostingowy, który wykorzystuje gradienty (pochodne funkcji kosztu) do trenowania kolejnych klasyfikatorów, a także:

- Shrinkage (learning rate)
- Regularizację
- Bagging w podpróbkowaniu obserwacji i cech.

◆ 4. XGBoost

✦ Idea

XGBoost (Extreme Gradient Boosting) to algorytm boostingowy, który wykorzystuje gradienty (pochodne funkcji kosztu) do trenowania kolejnych klasyfikatorów, a także:

- Shrinkage (learning rate)
 - Regularizację
 - Bagging w podpróbkiowaniu obserwacji i cech.
-

◆ 5. Sztuczne dane

Generujemy dane:

- $X_{ij} \sim N(0, 1)$ dla $i = 1, \dots, n, j = 1, \dots, 10$.
- Obliczamy sumę kwadratów zmiennych:

$$S_i = \sum_{j=1}^{10} X_{ij}^2$$

- Porównujemy S_i z medianą rozkładu chi-kwadrat o 10 stopniach swobody ($\chi_{10}^2(0.5)$):
 - $y_i = 1$ jeśli $S_i > \chi_{10}^2(0.5)$
 - $y_i = -1$ w przeciwnym razie.
-

◆ 6. Ewaluacja modeli

- Generujemy zbiory treningowe (np. 2000) i testowe (np. 10000).
- Trenujemy modele:
 - Single Tree
 - Bagging
 - AdaBoost (nasza implementacja)
 - RandomForest
 - XGBoost
- Sprawdzamy błędy klasyfikacji:

$$\text{Error} = 1 - \text{accuracy}$$

- Rysujemy wykres błędu w funkcji liczby estymatorów:
 - $n = 1, 5, 10, 20, 40, 60, 80, 100$.