Ensemble methods

Christos Dimitrakakis

November 25, 2025

Outline

Ensemble methods

Bagging

Boosting

General idea

- Concept class Π (e.g. linear classifiers)
- Base learner $\lambda : \mathcal{D} \to \Pi$ (e.g. perceptron)

General Algorithm

At iteration b,

1. Generate data D_b from D.
2. Get base predictor $\pi_b = \lambda(D_b)$

At the end, combine predictors $\pi_1, \ldots, \pi_B$.

Ensemble methods

Bagging

Boosting

Bagging

Algorithm

- Input: Data D, bags B, base learner λ
- For $b = 1, \ldots, B$
  - Sample with replacement $D_b \sim \mathrm{Unif}(D)^N$
  - Obtain predictor $\pi_b = \lambda(D_b)$.
- Return $\{\pi_b\}$

Bagging

Algorithm

- Input: Data D, bags B, base learner λ
- For $b = 1, \ldots, B$
  - Sample with replacement $D_b \sim \mathrm{Unif}(D)^N$
  - Obtain predictor $\pi_b = \lambda(D_b)$.
- Return $\{\pi_b\}$

The bagged predictor

In the end, we just combine all the predictors:

$$\pi = f\left(\sum_{b=1}^{B} \pi_b\right)$$

Bagging classifiers

Classification setting

- Weak learner $\lambda : D \to \Pi$
- Base hypotheses $\pi_b : X \to \{-1, 1\}$

with

$$\pi_b = \lambda(D_b), \qquad D_b \sim D$$

- Aggregate hypothesis

$$\pi(x) = \text{sgn}\left(\sum_{b=1}^{B} \pi_b(x)\right)$$

Bagging classifiers

Classification setting

- Weak learner $\lambda : D \to \Pi$
- Base hypotheses $\pi_b : X \to \{-1, 1\}$

with

$$\pi_b = \lambda(D_b), \qquad D_b \sim D$$

- Aggregate hypothesis

$$\pi(x) = \text{sgn}\left(\sum_{b=1}^{B} \pi_b(x)\right)$$

PAC property

Choosing $B \in O(\ln(T))$ results in error bounded by $\tilde{O}(d/T)$ with high probability, for binary classification and VC dimension d
Bagging is an Optimal PAC learner, Larsen 2024.

Regression setting

- Weak learner $\lambda : D \to \Pi$
- Base hypotheses $\pi_b : X \to \{-1, 1\}$

with

$$\pi_b = \lambda(D_b), \qquad D_b \sim D$$

Regression setting

- Weak learner $\lambda : D \to \Pi$
- Base hypotheses $\pi_b : X \to \{-1, 1\}$

with

$$\pi_b = \lambda(D_b), \qquad D_b \sim D$$

- Aggregate hypothesis

$$\pi(x) = \frac{1}{B} \sum_{b=1}^{B} \pi_b(x)$$

Sub-sample-and-aggregate

Algorithm

- Input: Data D, number of experts B, base learner λ
- For $B = 1, \ldots, B$

– Sample without replacement $D_b \sim \mathrm{Unif}(D)$ – Obtain predictor $\pi_b = \lambda(D_b)$.

- Return $\{\pi_b\}$

Sub-sample-and-aggregate

Algorithm

- Input: Data D, number of experts B, base learner λ
- For $B = 1, \ldots, B$

– Sample without replacement $D_b \sim \mathrm{Unif}(D)$ – Obtain predictor $\pi_b = \lambda(D_b)$.

- Return $\{\pi_b\}$

The aggregated predictor

$$\pi = f\left(\pi_1, \ldots, \pi_k\right)$$

Random forests

- Same as bagged trees

Random forests

- Same as bagged trees

Random subset of features

In random forests a subset $\hat{p}$ of the features is randomly consider for splitting.

Rationale

- Make the splits more random
- Consequently, the trees will look very different.

Ensemble methods

Bagging

Boosting

Boosting regression trees

Build predictors sequentially, so that at iteration b, we try to improve performance on the hardest examples.

Algorithm

- Input: Learner λ, data $D = (X, y)$, learning rate $\eta > 0$, set $r = y$.
- For $b = 1, 2, \ldots, B$:
  - Fit $\pi_b = \lambda(D, r)$
  - Update the aggregate: $\pi = \pi + \eta\pi_b(x)$
  - Update residuals: $r_t = r_t - \eta\pi_b(x_t)$
- Output final model:

$$\pi = \sum_{b=1}^{B} \eta\pi_b$$

Hedge

Prediction with expert advice

This is an online prediction problem, where, at each step t:

- We pick a_t of N possible decisions
- We observe the loss $\ell_t(a)$ for all $a \in [N]$.
- We then suffer a loss $\ell_t(a_t)$
- Our goal is to minimise the total loss $\sum_{t=1}^{T} \ell_t(a_t)$

Hedge

Prediction with expert advice

This is an online prediction problem, where, at each step t:

- We pick a_t of N possible decisions
- We observe the loss $\ell_t(a)$ for all $a \in [N]$.
- We then suffer a loss $\ell_t(a_t)$
- Our goal is to minimise the total loss $\sum_{t=1}^{T} \ell_t(a_t)$

Algorithm

- Learning rate $\eta \in [0, 1]$, Weights $w_t \in [0, 1]^N$, Iterations $T > 0$
- For $t = 1, \ldots, T$
  1. Calculate allocation $p_t = \frac{w_t}{\sum_{i=1}^{N} w_{i,t}}$
  2. Selct action $a_t = i$ with probability $p_{t,i}$.
  3. Observe loss $\ell_t \in [0, 1]^N$ for all actions.
  4. Suffer loss ℓ_t.
  5. Update all weights $w_{t+1,i} = w_{t,i} \exp(\eta \ell_{i,t})$.

AdaBoost/SAMME

AdaBoost can be seen as a version of Hedge with a varying learning rate.

Algorithm

- Weighted learning algorithm λ.
- Data D.
- $w_{i,t} = 1/T$.
- For $b = 1, \ldots, B$
  1. Choose allocation $p_b = \frac{w_b}{\sum_{i=1}^{N} w_{b,i}}$
  2. Get $\pi_b = \lambda(D, p_b)$
  3. Calculate error for all examples: $\ell_{b,t} = |\pi_b(x_t) - y_t|$.
  4. Average error $\epsilon_b = \sum_{t=1}^{T} p_{b,t}\ell_{b,t}$.
  5. Let $\eta_b = \ln \frac{1-\epsilon_b}{\epsilon_b} + \ln(C-1)$.
  6. Set $w_{b+1,t} = w_{b,t} \exp\left(\eta_b \mathbb{I}\left\{c_t \neq \pi_b(x_t)\right\}\right)$.
- Output

$$\pi(x) = \arg\max_c \sum_{b=1}^{B} \eta_b \mathbb{I}\left\{\pi_b(x) = c\right\}.$$