The University of Queensland
AUSTRALIA
CREATE CHANGE

Machine Leaning

COMP4702/COMP7703

Prac 10

Amelia Qiu

https://github.com/Amelia-Tong/MachineLearning_COMP4702/blob/main

Bootstrap

The **bootstrap** method is a **resampling** technique used to estimate statistics on a population by sampling a dataset with **replacement**.

Process:

1. Draw multiple samples (with **replacement**) from a dataset.
2. Each sample is the **same size** as the original dataset.

Bagging

1. Take *B* different bootstrapped training sets:

$$D_1, D_2, \dots, D_B$$

2. Build a separate prediction model using each D(.):

$$\hat{f}_1(x), \hat{f}_2(x), \dots, \hat{f}_B(x)$$

3. Combine resulting predictions (e.g. average for regression, majority vote for classification)

Advantages

Bagging reduces the variance of the base model:

- Assume the output if the base model, d_j are iid, then

$$E[y] = E\left[\sum_j \frac{1}{L} d_j\right] = \frac{1}{L} L \cdot E[d_j] = E[d_j]$$

$$Var(y) = Var\left(\sum_j \frac{1}{L} d_j\right) = \frac{1}{L^2} Var\left(\sum_j d_j\right) = \frac{1}{L^2} L \cdot Var(d_j) = \frac{1}{L} Var(d_j)$$

- In practice, the gain is unlikely to be this good, because the base models are not iid

$$Var(y) = \frac{1}{L^2} Var\left(\sum_j d_j\right) = \frac{1}{L^2}\left[\sum_j Var(d_j) + 2\sum_j \sum_{i<j} Cov(d_i, d_j)\right]$$

Out-of-Bag Error

- On average, each bagged tree makes use of around two-thirds of the observations.

- The remaining observations not used to a given bagged tree are referred to as the out-of-bag (OOB) observations.

- We can predict the response for the observation using each of the trees in which that observation was OOB. This will yield around B/3 predictions for the observation.

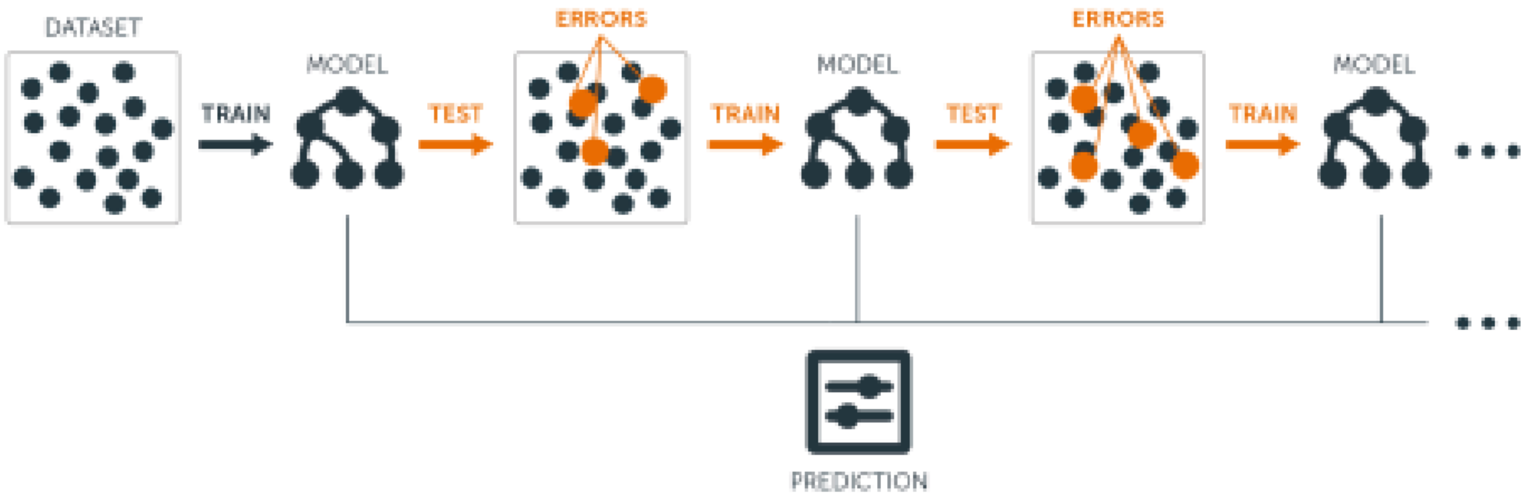
Random Forest

1. Take *B* different bootstrapped training sets.

2. Build *B* decision trees.
   (When building these trees, each time a split in a tree is considered, a random sample of m predictors is chosen as split candidates from the full set of p predictors.)

3. Combine resulting predictions (e.g. average for regression, majority vote for classification)

Boosting Tree



AdaBoost

Learn an AdaBoost classifier

Data: Training data $\mathcal{T} = \{\mathbf{x}_i, y_i\}_{i=1}^n$
Result: B weak classifiers

Assign weights $w_i^{(1)} = 1/n$ to all data points.
for $b = 1, \dots, B$ **do**
  Train a weak classifier $\widehat{y}^{(b)}(\mathbf{x})$ on the weighted training data $\{(\mathbf{x}_i, y_i, w_i^{(b)})\}_{i=1}^n$.
  Compute $E_{\text{train}}^{(b)} = \sum_{i=1}^n w_i^{(b)} \mathbb{I}\{y_i \neq \widehat{y}^{(b)}(\mathbf{x}_i)\}$.
  Compute $\alpha^{(b)} = 0.5 \ln((1 - E_{\text{train}}^{(b)})/E_{\text{train}}^{(b)})$.
  Compute $w_i^{(b+1)} = w_i^{(b)} \exp(-\alpha^{(b)} y_i \widehat{y}^{(b)}(\mathbf{x}_i))$, $i = 1, \dots, n$.
  Set $w_i^{(b+1)} \leftarrow w_i^{(b+1)} / \sum_{j=1}^n w_j^{(b+1)}$, for $i = 1, \dots, n$.
end

Predict with the AdaBoost classifier

Data: B weak classifiers with confidence values $\{\widehat{y}^{(b)}(\mathbf{x}), \alpha^{(b)}\}_{b=1}^B$ and test input $\mathbf{x}_\star$
Result: Prediction $\widehat{y}_{\text{boost}}^{(B)}(\mathbf{x}_\star)$

Output $\widehat{y}_{\text{boost}}^{(B)}(\mathbf{x}_\star) = \text{sign}\left\{\sum_{b=1}^B \alpha^{(b)} \widehat{y}^{(b)}(\mathbf{x}_\star)\right\}$.