Random Forest

Luu Minh Sao Khue

December 9, 2025

1. Introduction

Random Forest is an ensemble learning method that constructs a large collection of decision trees and combines their predictions through averaging (for regression) or majority vote (for classification). It is built on two key ideas:

- **Ensemble learning**, where multiple weak or moderately strong models are combined to obtain a more robust predictor.
- **Bagging** (Bootstrap Aggregating), which trains models independently on different bootstrap samples of the data to reduce variance.

Random Forest improves on bagging by adding *random feature selection*, which decorrelates the trees and makes the ensemble much stronger.

2. Ensemble Learning and Bagging

Ensemble learning combines the outputs of multiple models in order to reduce prediction variance and improve robustness. Decision trees are high-variance models, which makes them excellent candidates for ensemble methods.

Bagging (Bootstrap Aggregating)

Given training data $\{(x_i, y_i)\}_{i=1}^{N}$, bagging proceeds as follows:

1. Draw B bootstrap samples of size N by sampling with replacement.
2. Train one decision tree on each bootstrap sample.
3. Combine predictions of all trees:
    - **Regression:** average the predictions.
    - **Classification:** take the majority vote.

Bagging primarily reduces **variance**, which helps stabilize decision-tree–based models.

3. Random Forest: Bagging with Random Feature Selection

A Random Forest consists of many decision trees trained using bagging, with an additional randomization step:

> *At each split, instead of considering all features, a random subset of features is selected, and the best split is chosen only among those features.*

Formally, if the dataset has d features, then at each node the algorithm:

- Randomly selects m features ($m \ll d$).
- Computes the best split using impurity decrease (Gini/Entropy for classification, SSE/MSE for regression).

This random feature selection has two major advantages:

- Reduces tree-to-tree correlation.
- Strengthens the ensemble beyond bagging alone.

4. Random Forest Algorithm

Let B be the number of trees and m the number of selected features per split.

Algorithm 1 Random Forest (Classification or Regression)

```
Input: training data $\{(x_i, y_i)\}_{i=1}^{N}$, number of trees $B$, number of features per split $m$.
for $b = 1$ to $B$ do
    Draw bootstrap sample $D_b$ of size $N$ with replacement.
    Train a CART decision tree $T_b$ on $D_b$:
        • At each split, randomly choose $m$ features.
        • Select the best split using impurity decrease.
end for
Prediction:
    • Regression:
        $\hat{y}(x) = \frac{1}{B} \sum_{b=1}^{B} T_b(x).$
    • Classification:
        $\hat{y}(x) = \text{majority vote}\{T_1(x), \dots, T_B(x)\}.$
```

5. Advantages of Random Forest

Random Forest is widely used because it provides:

- Strong predictive performance with minimal tuning.
- Robustness to noise and outliers.
- Automatic handling of high-dimensional data.
- Built-in estimates of feature importance.
- Stability due to variance reduction.

6. Summary

Random Forest is a powerful ensemble of decision trees trained on bootstrap samples and randomized feature subsets. By combining bagging with random feature selection, it reduces both variance and correlation between trees, resulting in a stable, high-performing model suitable for many machine learning tasks.