Ensemble Methods: Bagging, Boosting, Random Forest, and Gradient Boosting

Prepared by Luu Minh Sao Khue

Introduction

Ensemble methods are powerful techniques in machine learning that combine the predictions of multiple models to improve accuracy and robustness. By aggregating predictions, ensemble methods reduce variance, bias, or improve predictions. Two primary types of ensemble methods are bagging and boosting.

1 Bagging vs Boosting

Bagging and boosting are two strategies for creating ensembles of models, but they differ fundamentally in how they construct and combine models.

1.1 Bagging (Bootstrap Aggregating)

Bagging works by training multiple models independently on different subsets of the training data and averaging their predictions.

- **Steps:**
  a) Generate m bootstrap samples (random samples with replacement) from the training dataset.
  b) Train a separate model on each bootstrap sample.
  c) Aggregate the predictions of all models (e.g., majority vote for classification or averaging for regression).
- **Goal:** Reduce variance and prevent overfitting.
- **Examples:** Random Forests.

1.2 Boosting

Boosting focuses on sequentially training models, where each subsequent model tries to correct the errors of the previous ones.

- **Steps:**
  a) Train the first model on the original dataset.
  b) Calculate errors and assign higher weights to the misclassified samples.
  c) Train the next model on the weighted dataset.
  d) Repeat until a stopping criterion is met (e.g., number of models, error threshold).
  e) Combine the models' predictions, often using a weighted majority vote or sum.
- **Goal:** Reduce bias and improve accuracy.
- **Examples:** AdaBoost, Gradient Boosting.

2 Random Forests

Random Forest is an ensemble learning method based on bagging, which builds multiple decision trees and aggregates their predictions.

2.1 Steps:

a) Generate m bootstrap samples (random samples with replacement) from the training dataset.

b) For each sample, build a decision tree using a random subset of features at each split:

   Gini Index: G = 1 - $\sum_{k=1}^{K} p_k^2$, where p_k is the proportion of samples of class k at a node.

c) Aggregate the predictions of all trees:
   - For classification: Use majority voting.
   - For regression: Take the average of the predictions.

2.2 Advantages and Disadvantages

- **Advantages:**
  - Reduces overfitting compared to a single decision tree.
  - Handles high-dimensional data effectively.
  - Provides feature importance scores.
- **Disadvantages:**
  - Can be computationally expensive for large datasets.
  - Less interpretable than a single decision tree.

2.3 Applications

Random Forests are widely used in classification, regression, and feature selection tasks.

3 AdaBoost (Adaptive Boosting)

AdaBoost builds an ensemble by assigning weights to data points and iteratively focusing on harder examples.

3.1 Steps:

a) Assign initial weights $w_i = \frac{1}{n}$ to all n training samples.

b) For each iteration $t = 1, 2, \ldots, T$:

   i. Train a weak learner $h_t(x)$ on the weighted dataset.

   ii. Compute the weighted error:

$$\varepsilon_t = \frac{\sum_{i=1}^{n} w_i \mathbb{I}(h_t(x_i) \neq y_i)}{\sum_{i=1}^{n} w_i},$$

   where $\mathbb{I}$ is the indicator function.

   iii. Calculate the model weight:

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \varepsilon_t}{\varepsilon_t} \right).$$

   iv. Update sample weights:

$$w_i \leftarrow w_i \exp(-\alpha_t y_i h_t(x_i)),$$

   and normalize so that $\sum w_i = 1$.

Final prediction is the weighted majority vote:

$$H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right).$$

3.2 Advantages and Disadvantages

- **Advantages:** Simple, effective for weak learners, and robust to overfitting.
- **Disadvantages:** Sensitive to noisy data and outliers.

4 Gradient Boosting

Gradient Boosting optimizes an objective function by training models sequentially to minimize the residual errors of the previous models.

4.1 Steps:

a) Initialize the model with a constant prediction:

$$F_0(x) = \arg\min_{\gamma} \sum_{i=1}^{n} L(y_i, \gamma),$$

where L is the loss function (e.g., mean squared error for regression).

b) For each iteration $t = 1, 2, \ldots, T$:

   i. Compute the residuals (pseudo-residuals):

$$r_{i,t} = -\frac{\partial L(y_i, F_{t-1}(x_i))}{\partial F_{t-1}(x_i)}.$$

   ii. Train a weak learner $h_t(x)$ to predict the residuals $r_{i,t}$.

   iii. Update the model:

$$F_t(x) = F_{t-1}(x) + \eta h_t(x),$$

   where η is the learning rate.

Final prediction is:

$$F_T(x) = F_0(x) + \sum_{t=1}^{T} \eta h_t(x).$$

4.2 Advantages and Disadvantages

- **Advantages:** Handles complex relationships, flexible with custom loss functions, and provides state-of-the-art results for many tasks.

- **Disadvantages:** Computationally expensive, sensitive to hyperparameters, and prone to overfitting if not regularized.

5 Comparison of Ensemble Methods

Feature	**Random Forest**	**AdaBoost**	**Gradient**
Base Learner	Decision Trees	Weak Learners	Weak L
Training Style	Parallel (Bagging)	Sequential (Boosting)	Sequential
Goal	Reduce Variance	Reduce Bias	Optimize Obje
Robustness to Noise	High	Low	Mod
Hyperparameter Sensitivity	Low	Moderate	Hi
Computational Efficiency	High	Moderate	Lo
Applications	Feature Selection, General ML	Classification Tasks	Regression, C

Table 1: Comparison of Ensemble Methods