Gradient Boosting: From Basics to Advanced Implementations

Arnav Samal

August 10, 2024

1 Introduction to Gradient Boosting

Gradient Boosting is a powerful machine learning technique that creates a strong predictive model by combining multiple weak learners, typically decision trees.

Definition 1. *Gradient Boosting is an ensemble learning method that builds a series of weak learners (usually decision trees) sequentially, with each new model trying to correct the errors of the combined previous models.*

2 Basic Gradient Boosting Algorithm

Algorithm 1 Basic Gradient Boosting

procedure GradientBoosting($F_0, L, \{(x_i, y_i)\}_{i=1}^n, M$)
  Initialize $F_0(x) = \arg\min_\gamma \sum_{i=1}^n L(y_i, \gamma)$
  for $m = 1$ to M **do**
    Compute pseudo-residuals:
    $r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)}$ for $i = 1, \ldots, n$
    Fit a weak learner (e.g., decision tree) $h_m(x)$ to pseudo-residuals
    Compute multiplier $\gamma_m = \arg\min_\gamma \sum_{i=1}^n L(y_i, F_{m-1}(x_i) + \gamma h_m(x_i))$
    Update model: $F_m(x) = F_{m-1}(x) + \gamma_m h_m(x)$
  end for
  return $F_M(x)$
end procedure

Where:

- F_0: Initial model
- L: Loss function
- (x_i, y_i): Training data

- M: Number of iterations
- h_m: Weak learner at iteration m
- γ_m: Step size at iteration m

3 Gradient Boosting with Regularization

To prevent overfitting, we introduce regularization terms:

$$F_m(x) = F_{m-1}(x) + \nu\gamma_m h_m(x) \tag{1}$$

where ν ($0 < \nu \leq 1$) is the learning rate.

4 Common Variants

4.1 XGBoost (Extreme Gradient Boosting)

XGBoost introduces several improvements:

1. Regularized objective:

$$\text{Obj} = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{2}$$

   where $\Omega(f) = \gamma T + \frac{1}{2}\lambda||w||^2$

2. Approximate split finding for faster training
3. Handling sparse data
4. Built-in cross-validation

4.2 LightGBM (Light Gradient Boosting Machine)

LightGBM introduces:

1. Gradient-based One-Side Sampling (GOSS)
2. Exclusive Feature Bundling (EFB)
3. Leaf-wise tree growth instead of level-wise

4.3 GBM (Gradient Boosting Machine)

GBM refers to the original implementation of Gradient Boosting, available in popular libraries like scikit-learn. It typically offers:

1. Basic boosting with configurable loss functions
2. Support for different base estimators
3. Built-in cross-validation and hyperparameter tuning

4.4 CatBoost

CatBoost (Categorical Boosting) introduces:

1. Efficient handling of categorical features without the need for one-hot encoding
2. Ordered boosting, which reduces prediction shift and overfitting
3. GPU support for faster training
4. Built-in handling of missing values

5 Comparison of Implementations

Feature	Basic GB	XGBoost	LightGBM	CatBoost
Speed	Slow	Fast	Very Fast	Fast
Memory Usage	High	Medium	Low	Medium
Accuracy	Good	Very Good	Very Good	Excellent
Handling Categorical Variables	Poor	Good	Excellent	Excellent

Table 1: Comparison of Gradient Boosting Implementations

6 Mathematical Foundations

6.1 Gradient Descent in Function Space

Gradient Boosting can be viewed as gradient descent in function space. At each iteration m, we update our model:

$$F_m(x) = F_{m-1}(x) - \rho_m \nabla_F L(y, F_{m-1}(x)) \tag{3}$$

Where ρ_m is the step size and $\nabla_F L$ is the functional gradient.

6.2 Loss Functions

Common loss functions include:

- Squared Error (Regression): $L(y, F) = \frac{1}{2}(y - F)^2$
- Logarithmic Loss (Classification): $L(y, F) = \log(1 + e^{-2yF})$
- Absolute Error (Robust Regression): $L(y, F) = |y - F|$

7 Advanced Topics

7.1 Feature Importance

Feature importance in Gradient Boosting is often calculated based on the number of times a feature is used to split the data across all trees, weighted by the improvement in the model as a result of each split.

7.2 Handling Imbalanced Data

Techniques for handling imbalanced data include:

- Adjusting class weights
- Focal loss
- Sampling techniques (oversampling, undersampling)

7.3 Hyperparameter Tuning

Key hyperparameters to tune:

- Number of estimators
- Learning rate
- Maximum tree depth
- Minimum samples per leaf
- Subsample ratio

8 Conclusion

Gradient Boosting, especially in its modern implementations like XGBoost, LightGBM, GBM, and CatBoost, represents a powerful and flexible approach to machine learning. Its ability to handle various types of data and problems, combined with high predictive accuracy, makes it a popular choice in many applications, from Kaggle competitions to real-world industry problems.