Gradient Boosting for Stock Price Prediction

Mangukiya Raj
Department of Mathematics
Sardar Vallabhbhai National Institute of Technology, Surat

April 21, 2025

1. Introduction

Gradient Boosting is a powerful and widely used machine learning technique that aims to improve the predictive performance of a model by building an ensemble of weak learners in a sequential manner. It is most commonly used for regression and classification tasks. The key idea is to combine multiple models such that each new model focuses on the mistakes made by the previous ones, thereby gradually reducing the overall error.

In stock price prediction, Gradient Boosting has proven effective due to its ability to capture complex patterns in time-series data. Unlike linear models, it does not assume a fixed form for the data relationship, making it well-suited for financial data which is often noisy, non-linear, and affected by many variables.

2. Working of Gradient Boosting

Gradient Boosting works by starting with an initial simple prediction and then iteratively improving it. Each iteration involves fitting a new model to the residuals (errors) of the current model, thus reducing the overall loss function step by step. This is done using gradient descent, which is a mathematical optimization technique.

Let the training data be $\{(x_i, y_i)\}_{i=1}^n$, where x_i are input features and y_i are the true outputs. The model starts with an initial guess:

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

where $L(y, \hat{y})$ is a differentiable loss function (e.g., Mean Squared Error).

At the m-th iteration, the model computes the negative gradient (pseudo-residuals):

$$r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F=F_{m-1}}$$

Then it fits a weak learner (usually a decision tree) $h_m(x)$ to these residuals. The output is updated as:

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

where η is the learning rate, and γ_m is the step size found by minimizing:

$$\gamma_m = \arg\min_{\gamma} \sum_{i=1}^{n} L(y_i, F_{m-1}(x_i) + \gamma h_m(x_i))$$

3. Mathematical Foundation and Definitions

To better understand Gradient Boosting, it's important to define some key mathematical terms:

Loss Function (L): A function that measures the error between the predicted value $\hat{y}$ and the actual value y. For regression problems, a common choice is Mean Squared Error (MSE):

$$L(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Gradient: In optimization, a gradient represents the direction and rate of fastest increase of a function. In Gradient Boosting, we compute the negative gradient to determine the direction to reduce the loss.

Residuals: These are the differences between the actual output and the model's predictions. Gradient Boosting treats these residuals as the target for the next learner.

Learning Rate (η): A small constant used to control how much the model is adjusted at each step. Smaller values lead to more accurate but slower training.

Base Learner ($h_m(x)$): Usually a decision tree with limited depth, trained to predict the residuals.

Ensemble Model: A final model made by combining several weak learners. The output is the sum of their weighted predictions.

Gradient Boosting minimizes the loss function iteratively by adding a new base learner that predicts the gradient of the loss with respect to the current model prediction.

4. Gradient Boosting in Stock Price Prediction

Stock price prediction is a challenging task due to the non-stationary and volatile nature of financial time series. Gradient Boosting is well-suited for this task because it:

- Does not assume linear relationships between variables
- Can capture non-linear and complex interactions

- Reduces overfitting when regularization is applied
- Handles missing data and noisy patterns efficiently

In practice, historical stock data (like past prices, moving averages, RSI, and other indicators) are used as input features. The output is the predicted price for the next time step. Gradient Boosting learns from these patterns and corrects its mistakes in each round, improving prediction accuracy.

5. Example

Consider a dataset where the goal is to predict the next day's closing price based on the past 5 days. Let:

$$X = \begin{bmatrix} 150 \\ 152 \\ 149 \\ 153 \end{bmatrix}, \quad y = \begin{bmatrix} 152 \\ 149 \\ 153 \\ 155 \end{bmatrix}$$

A weak model may first predict $\hat{y}^{(1)} = [151, 150, 151, 152]$, resulting in residuals:

$$r = y - \hat{y}^{(1)} = [1, -1, 2, 3]$$

The next model fits to these residuals and predicts adjustments. After a few rounds, the combined prediction becomes more accurate:

$$\hat{y}^{\text{final}} = \hat{y}^{(1)} + \text{adjustments}$$

This process continues until a stopping criterion (number of trees or minimum loss) is met.

6. Key Hyperparameters in Gradient Boosting

Tuning hyperparameters is essential for optimizing model performance and avoiding overfitting. The most important ones include:

- **Learning Rate** (η): Controls the contribution of each tree. Lower values make the model more robust.
- **Number of Trees** (M): More trees improve learning but may lead to overfitting.
- **Tree Depth**: Determines model complexity. Shallow trees are preferred for faster computation.
- **Subsample**: Fraction of data used for training each tree. Helps reduce variance.

7. Variants and Extensions

Several advanced versions of Gradient Boosting have been developed to improve speed and accuracy:

- **XGBoost**: Regularized version of Gradient Boosting that is faster and often more accurate.
- **LightGBM**: Designed for large datasets with faster training and lower memory usage.
- **CatBoost**: Handles categorical variables efficiently.

These variants are widely used in competitive data science and real-world applications.

8. Advantages and Limitations

Advantages:

- High prediction accuracy
- Works well on small to medium-sized datasets
- Handles mixed data types (categorical and continuous)
- Offers feature importance insights

Limitations:

- Computationally intensive
- Sensitive to hyperparameters (learning rate, number of trees)
- Can overfit if not properly tuned

9. Real-World Use Cases

Gradient Boosting is widely used in the real world, particularly in finance, healthcare, marketing, and e-commerce. In healthcare, it is used for disease prediction and medical diagnostics. In marketing, it helps in customer segmentation and churn prediction. In e-commerce, it powers product recommendations and personalization engines.

In the financial sector, Gradient Boosting is employed in credit scoring, fraud detection, portfolio optimization, and time-series forecasting. Its flexibility and high accuracy make it a preferred algorithm in various data-driven applications across different industries.

10. Application in Stock Markets

A specific application of Gradient Boosting in finance includes predicting the next-day closing price of major stock indices such as the S&P 500, NASDAQ, or specific company stocks like Apple or Tesla. Historical stock price data (Open, High, Low, Close, Volume), along with technical indicators such as Moving Average, Bollinger Bands, RSI (Relative Strength Index), and MACD (Moving Average Convergence Divergence), are used as input features.

In such a model, the data pipeline includes preprocessing (handling missing values, feature scaling), feature engineering (creating lag variables), training using Gradient Boosting (e.g., with XGBoost or LightGBM), and then evaluation using metrics such as RMSE or MAPE. Once trained, the model is able to learn from previous patterns and make accurate short-term price predictions, which can then be used to inform investment decisions.

In algorithmic trading, Gradient Boosting is used in combination with backtesting strategies, where predicted values guide the buy or sell decisions of a simulated or real-time trading bot. This practical use of Gradient Boosting demonstrates its ability to outperform traditional statistical models, especially when the underlying financial signals are noisy and non-linear.

10. References

- Friedman, J. H. (2001). "Greedy Function Approximation: A Gradient Boosting Machine." The Annals of Statistics.
- Chen, T., & Guestrin, C. (2016). "XGBoost: A Scalable Tree Boosting System." KDD Conference.
- https://scikit-learn.org/stable/modules/ensemble.html#gradient-boosting
- https://xgboost.readthedocs.io/en/stable/
- https://www.investopedia.com/terms/g/gradient-boosting.asp