Advancements and Applications of Gradient Boosting: A Comprehensive Literature Review

Akshay Merugu
University of Alabama at Birmingham
dept. of CS
Birmingham, AL
amerugu@uab.edu

Venkatesh Jella
University of Alabama at Birmingham
dept. of CS)
Birmingham, A
vjella@uab.edu

Rahul Bollepalli
University of Alabama at Birmingham
dept. of CS
Birimingham, AL
rbollepa@uab.edu

Jaydeep Chaudary
University of Alabama at Birmingham
dept. of CS
Birmingham, AL
jchaudhary@uab.edu

I. Introduction

In recent years, gradient boosting has emerged to be a powerful ensemble learning technique that is widely recognized for its results in regression and classification tasks across various domains such as finance, healthcare, banking and natural language processing. Gradient boosting is originally introduced to address the limitations and weaker results provided by weaker/ traditional learning methods, by minimizing the loss function and improving the accuracy through the addition of weaker models together.

The evolution of gradient boosting from early algorithms like ADABoost to advanced methods such as XGBoost, LightGBM, CatBoost has significantly improved efficiency and the ability to handle large datasets, which are pretty crucial in this era of big data.

II. History

Schapire and Freund's AdaBoost (1996): In 1996, Yoav Freund and Robert Schapire introduced the AdaBoost (Adaptive Boosting) algorithm, which was a breakthrough in ensemble methods. AdaBoost worked by adjusting the weights of misclassified samples, allowing the model to "focus" on harder cases in each iteration. This method was highly influential and demonstrated the power of boosting in improving prediction accuracy. Freund and Schapire received the Gödel Prize in 2003 for their work on AdaBoost.

Introduction of Gradient Boosting (1999) Jerome Friedman's Contribution: The concept of gradient boosting was formalized by Jerome H. Friedman in a seminal paper published in 1999 titled "Greedy Function Approximation: A Gradient Boosting Machine." Friedman's work expanded on the principles of boosting by introducing a gradient-based approach. Instead of focusing on misclassified samples, gradient boosting minimizes a loss function (e.g., mean squared error or log loss) through gradient descent in a function space. This allows for more flexibility, as it can optimize any differentiable loss function rather than being limited to classification errors. Gradient Boosting Framework: Friedman's framework made it possible to apply boosting to a variety of loss functions, enabling its use in both regression and classification tasks. This flexibility and adaptability set gradient boosting apart from AdaBoost and solidified it as a powerful machine-learning method.

III. Theoretical Background

As renowned professor ChengLi from NEU in his lectures (A Gentle Introduction to Gradient Boosting) , states that Gradient Boosting is a combination of Gradient Descent and Boosting. Wherein, Gradient Descent is a unique method which is used for optimization, usually during the training phase of the machine learning model. It primarily emphasizes on a function called convex function, where it adjusts the parameters in a step-by-step fashion in order to reduce a function to its local minimum. An easier and simpler way to understand a gradient would be that the model can learn at the highest speed possible when the slope of the function is as steep as it can get. $\mathbf{b} = \mathbf{a} - \gamma \nabla \mathbf{f}(\mathbf{a})$ Where, The following implementation of the equation shows the general procedure of the GD algorithm. In this illustration b indicates the next position of the climber, whereas a the current. the "-" sign symbolises the minimization . The gamma symbol acts as a learning rate and the term f(a) points the direction of the descent where it is the steepest.

A. *Boosting*

"The term 'Boosting' refers to a family of algorithms which converts weak learners to strong learners." as stated by Sunil Ray in his article from With the help of varied distributions, we identify the weak learners by the application of base learning or machine learning. In the Iteration of the algorithm, with every iteration a new weak prediction is created. after few of these iterations, all of these weak predictions are merged and combined into a strong, firm and independent prediction rule.

The steps for the selection for an optimal distribution are :

- initial distribution : In this step equal weights are assigned to each observation and all the distributions are considered by the base learner.
- Adjusting for errors: Let's say there are errors in the prediction of the initial base, then we provide it with higher attention , further where the next base algorithm is applied
- Iteration: The previous step keeps on repeating until the algorithm reaches the highest

In the Final step, all the outputs from the previous weak learners are combied together in order to form a strong, and higher accuracy yielding learner. Which further enhances the model's prediction accuracy.

As Cheng Li explains, gradient boosting can be generalized by accumulating a unique loss function and its corresponding gradient, this is a concept which he demonstrates mathematically in one of his lectures.

Like other boosting methods, gradient boosting combines weak "learners" into a single strong learner iteratively It is easiest to explain in the least-squares regression setting, where the goal is to "teach" a model "F" to predict values of the form

$$\hat{y} = F(x)$$

by minimizing the mean squared error

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

where i indexes over some training set of size n of actual values of the output variable y:

$$\begin{aligned} \hat{y}_i &= F(x_i) \quad \text{(the predicted value)} \\ y_i &= \text{(the observed value)} \\ n &= \text{(the number of samples in } y\text{)} \end{aligned}$$

Consider an algorithm with M stages. At each stage m ($1 \leq m \leq M$), let's assume an imperfect model F_m is generated (for smaller values of m, this model might simply predict $\hat{y}_i = \bar{y}$, the mean of y). To improve F_m, the algorithm adds a new estimator $h_m(x)$, such that:

$$F_{m+1}(x_i) = F_m(x_i) + h_m(x_i) = y_i$$

This can also be expressed as:

$$h_m(x_i) = y_i - F_m(x_i).$$

Consequently, gradient boosting fits h_m to the residuals $y_i - F_m(x_i)$. Like other boosting methods, each new model F_{m+1} aims to correct the errors of the prior model F_m. Extending this concept to different loss functions (beyond squared error), as well as to classification and ranking problems, comes from noting that the residuals $h_m(x_i)$ are proportional to the negative gradients of the mean squared error (MSE) loss function with respect to $F(x_i)$:

$$L_{\text{MSE}} = \frac{1}{n}\sum_{i=1}^{n}(y_i - F(x_i))^2$$

$$-\frac{\partial L_{\text{MSE}}}{\partial F(x_i)} = \frac{2}{n}(y_i - F(x_i)) = \frac{2}{n}h_m(x_i).$$

Thus, gradient boosting can be adapted into a gradient descent approach by integrating an alternative loss function and its corresponding gradient.

IV. Evolution Of Gradient Boosting

A. *Foundational Papers/Studies*

The First series of studies on gradient boosting were published by Friedman, J. H in 2001 and 2002 : The first study focused on the gradient boosting, providing the theoretical basis and describing its application as an optimization algorithm in function space. The second study focused on the stochastic gradient boosting, adding randomization to enhance generalizability and reduce overfitting.

B. *Key Insights*

- Boosting as Iterative, Additive Model: The capability of gradient boosting to iteratively construct models and reduce residuals results in systematic enhancement of the overall prediction accuracy.
- Gradient Descent in Function Space: Accomplishing gradient descent on loss in function space offers flexibility and generality for a wide range of tasks.
- Stochastic Gradient Boosting: It introduces randomness in order to reduce overfitting, hence giving better generalization. Regularization Techniques: Some of the techniques include reducing
- Feature Importance Foundation: Laying a foundation for feature importance, which shall help in enhancing interpretability in machine learning models.

These insights from Friedman's foundational work have influenced the latest versions of Gradient Boosting-to wit, XGBoost, LightGBM, and CatBoost-which implement these principles with various other optimizations in scalability, efficiency, and complex data management.

C. *XGBoost*

Tianqi Chen's research project within the Distributed ML community has pathed its way into the XGBoost. This was initially developed as an application which can only be used with the terminal which can only be configured with a libsvm file, it later gained its recognition in various ML competitions, especially after it bagging the higgs ML challenge. This immense success led to the expansion of this algorithm to be included into various languages such as python, R, Java, etc. and this became so popular with developers that this was widely used in many competitions.

D. *Studies on XGBoost*

The Foundational study on XGBoost was " XGBoost: A Scalable Tree Boosting System Tianqi Chen, Carlos Guestrin"

There research's conclusion says " In this paper, we described the lessons we learnt when building XGBoost, a scalable tree boosting system that is widely used by data scientists and provides state-of-the-art results on many problems. We proposed a novel sparsity aware algorithm for handling sparse data and a theoretically justified weighted quantile sketch for approximate learning. Our experience shows that cache access patterns, data compression and sharding are essential elements for building a scalable end-to-end system for tree boosting. These lessons can be applied to other machine learning systems as well. By combining these insights, XGBoost is able to solve real world scale problems using a minimal amount of resources"

E. *Key Insights on XGBoost*

The paper "XGBoost: A Scalable Tree Boosting System" by Chen and Guestrin introduces XGBoost, an efficient implementation of gradient boosting aimed at improving computational speed and performance and resolving several limitations of previous algorithms. Key features include L1 and L2 regularization techniques to prevent overfitting, a sparsity-aware algorithm which copes well with missing values, and the opportunity for parallel processing while constructing trees, leading to dramatic reductions in training time. This weighted quantile sketch algorithm introduces efficient split finding in large-scale datasets, while the built-in cross-validation and early stop mechanisms provide optimized hyperparameter tuning and model evaluation. XGBoost allows various objective functions, hence one can tune it for a specific task; shows strong empirical performance, beating competitors on many machine learning benchmarks. This is because of its flexibility, scalability, and also the robustness of its features, which have made its wide adoption in data science and also place it as the cornerstone of most machine learning applications across wide domains.

F. *Studies on LightGBM*

The most significant study on LightGBM is a research made by the Microsoft Research team, which consists of Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, Tie-Yan Liu

G. *Key Insights on LightGBM*

Their paper states that " we have proposed a novel GBDT algorithm called LightGBM, which contains two novel techniques: Gradient-based One-Side Sampling and Exclusive Feature Bundling to deal with large number of data instances and large number of features respectively. We have performed both theoretical analysis and experimental studies on these two techniques. The experimental results are consistent with the theory and show that with the help of GOSS and EFB, LightGBM can significantly outperform XGBoost and SGB in terms of computational speed and memory consumption. For the future work, we will study the optimal selection of a and b in Gradient-based One-Side Sampling and continue improving the performance of Exclusive Feature Bundling to deal with large number of features no matter they are sparse or not. "

H. *Impact of LightGBM on MachineLearning*

LightGBM's innovations in memory efficiency, computational speed, and handling of large datasets have positioned it as a go-to tool in machine learning, especially in applications requiring fast, scalable, and accurate models

V. Application Based Studies on XGBoost

A. *XGBoost for morality Prediction*

Chen, W., Liu, C., Peng, L., He, H., Su, J. (2020). "Using XGBoost Algorithm and SHAP for Feature Importance Analysis in Predicting Mortality of COVID-19 Patients."

- This study used XGBoost to predict patient outcomes and applied SHAP values to explain feature importance, demonstrating how XGBoost can provide interpretable results in healthcare.

B. *Share Market Forecasting in Chinese market*

Li, Y., Dai, Y., Wang, H., Chen, W. (2020). "Forecasting Stock Market Movements with Ensemble Learning Methods: An Empirical Study in Chinese A-Share Market."

- This study explores XGBoost in financial prediction, noting its strengths in handling complex and volatile financial data for improved market forecasting.

VI. Future Trends and Directions

A. *Enhanced Interpretability and Explainability*

As gradient boosting models are often used in high-stakes domains (e.g., finance, healthcare), developing more transparent models will be crucial. Techniques such as SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-Agnostic Explanations) are likely to be refined further Lundberg, S. M., Lee, S.-I. (2017). "A Unified Approach to Interpreting Model Predictions." Advances in Neural Information Processing Systems (NeurIPS).

- This foundational paper introduces SHAP values for model interpretability, including gradient boosting models. SHAP has become widely used for explaining predictions and improving model transparency.

B. *Incorporation of Neural Network Components*

Recent research has explored hybrid models that combine the strengths of gradient boosting and deep learning, such as deep neural decision forests. Kontschieder, P., Fiterau, M., Criminisi, A., Bulo, S. R. (2015). "Deep Neural Decision Forests." Proceedings of the IEEE International Conference on Computer Vision (ICCV).

- This paper presents a hybrid model combining decision trees with neural networks, a concept that has informed the exploration of blending gradient boosting with neural network layers

C. *Automatic Hyperparameter Tuning and Meta-Learning*

AutoML (Automated Machine Learning) has gained momentum, and the next generation of gradient boosting frameworks may come with built-in automated hyperparameter tuning or meta-learning capabilities, allowing models to automatically adapt hyperparameters to the dataset's specific characteristics without manual intervention.

Feurer, M., Hutter, F. (2019). "Hyperparameter Optimization." Automated Machine Learning, 3-33.

- A detailed overview of hyperparameter optimization techniques, discussing their application in boosting frameworks and AutoML, which will play a role in the evolution of automated tuning in gradient boosting.

VII. Conclusion

In conclusion, Gradient boosting has evolved from a foundational technique to a highly optimzed framework that is extensively used in machine learning across various domains From foundational techniques like AdaBoost to innovative frameworks such as XGBoost, LightGBM, and CatBoost, such developments have influenced machine learning in terms of computational efficiency and also broadened the range of tasks for which gradient boosting applies.

Moving forward, the trends in improved interpretability, integration with neural networks, and automated hyperparameter tuning are showing a promising future for gradient boosting. These developments make gradient boosting more accessible and applicable to complex, large-scale datasets, further solidifying its leading position both in research and industry oriented programs.

References

[1] Friedman, J. H. (2001). "Greedy Function Approximation: A Gradient Boosting Machine." Annals of Statistics, 29(5), 1189-1232.

[2] Friedman, J. H. (2002). "Stochastic Gradient Boosting." Computational Statistics and Data Analysis, 38(4), 367-378.

[3] Chen, T., and Guestrin, C. (2016). "XGBoost: A Scalable Tree Boosting System." Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785-794.

[4] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., and Liu, T.-Y. (2017). "LightGBM: A Highly Efficient Gradient Boosting Decision Tree." Advances in Neural Information Processing Systems, 30, 3146-3154.

[5] Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., and Gulin, A. (2018). "CatBoost: Unbiased Boosting with Categorical Features." Advances in Neural Information Processing Systems, 31, 6638-6648.

[6] Natekin, A., and Knoll, A. (2013). "Gradient Boosting Machines, a Tutorial." Frontiers in Neurorobotics, 7, 21.

[7] Zhang, C., and Ma, Y. (2021). "Gradient Boosting Machines: A Survey and Review." IEEE Transactions on Knowledge and Data Engineering, 33(1), 1-17.

[8] Dey, K., Saha, P., and Chaudhuri, S. (2017). "Application of Gradient Boosting Machine in Healthcare: Prediction of Risk for Heart Failure Using Data from the Framingham Heart Study." Healthcare Informatics Research, 23(2), 121-129.

[9] Li, Y., Dai, Y., Wang, H., and Chen, W. (2020). "Forecasting Stock Market Movements with Ensemble Learning Methods: An Empirical Study in Chinese A-Share Market." Expert Systems with Applications, 160, 113704.

[10] https://www.analyticsvidhya.com/blog/2015/11/quick-introduction-boosting-algorithms-machine-learning

[11] https://www.chengli.io/tutorials/gradient_boosting.pdf

[12] Lundberg, S. M., and Lee, S.-I. (2017). "A Unified Approach to Interpreting Model Predictions." Advances in Neural Information Processing Systems (NeurIPS).

[13] Chen, J., Song, L., Wainwright, M. J., and Jordan, M. I. (2020). "Learning to Explain: An Information-Theoretic Perspective on Model Interpretation." Proceedings of the 37th International Conference on Machine Learning (ICML).

[14] Popov, S., Morozov, V., and Babenko, A. (2019). "Neural Oblivious Decision Ensembles for Deep Learning on Tabular Data." International Conference on Learning Representations (ICLR).

[15] Strubell, E., Ganesh, A., and McCallum, A. (2019). "Energy and Policy Considerations for Deep Learning in NLP." Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics.

[16] Yu, J., Ko, S., and Lee, C. (2020). "Energy-Efficient Gradient Boosting Decision Tree for Embedded Devices." IEEE Access, 8, 70882-70894.

.