MAXGBoost: A Fast Heuristic Approach to Adaptive Learning Rates in Gradient Boosted Decision Trees

Anirudh Chintaluri[1], Vini Rezanejad[2]

[1] Anirudh Chintaluri, anirudh.chintaluri@gmail.com, Thomas Jefferson High School for Science and Technology, 6560 Braddock Rd, Alexandria, VA 22312

[2] Vini Rezanejad, radin.rezanezhad@gmail.com, Thomas Jefferson High School for Science and Technology, 6560 Braddock Rd, Alexandria, VA 22312

*Both authors contributed equally to this research

1. Abstract

Credit card fraud detection presents unique challenges due to highly complex imbalanced datasets and the need for efficient model training while maintaining high performance. This paper introduces Momentum Approximation XGBoost (MAXGBoost), a novel approach that dynamically adjusts gradient boosted decision tree (GBDT) learning rates (η) depending on the loss landscape. Unlike traditional GBDTs that use static or predetermined decay schedules for η, MAXGBoost adapts η based on the loss momentum, reducing the need for hyperparameter tuning while improving convergence speed. We evaluated our approach against Decision Tree, Random Forest, constant η XGBoost, and exponential decay η XGBoost implementations on a real-world credit card transaction dataset containing 284,807 transactions with only 0.2% fraudulent cases. Results demonstrate that MAXGBoost achieves superior precision (0.93827) and accuracy (0.99980) compared to other models, though the exponential decay learning rate approach showed the best recall (0.97531) and AUC (0.98757). Our findings indicate that incorporating momentum approximation η adaptation in GBDTs provides a promising framework for robust fraud detection capabilities.

2. Introduction

Fundamentally, Gradient Boosted Decision Trees (GBDTs) work by initializing a decision tree that will be evaluated on a metric, known as the loss function. After analyzing the loss of the initial decision tree, the model will attempt to minimize the loss by moving in the direction of the negative of the gradient — that is, move in the direction where the loss function is decreasing the fastest so that the loss can be minimized and the model performance can be maximized. This behaves very similarly to neural networks' backpropagation algorithm with the multivariate correction techniques but instead involves a decision tree that eventually boosts itself to form an ensemble decision tree model.

In order for this GBDT algorithm to apply the direction of the negative of the gradient to the decision tree, it will add another decision tree scaled by a factor called the learning rate (η), showing how much the tree will go down the loss function. The choice of η is critical. A low η ensures convergence but requires more iterations, increasing computational costs. Conversely, a high ηaccelerates convergence but risks overshooting the optimal solution (Baranovskij, 2019). According to Brownlee (2016), tuning η in GBDTs often requires iterative experimentation, as the optimal η depends on both the dataset and the specific loss function. A well-tuned η schedule can reduce the reliance on extensive hyperparameter grid searches, which are computationally expensive. Moreover, Baranovskij (2019) highlights that adaptive strategies can reduce the sensitivity to initial η choices, particularly in datasets with high sparsity, where the gradient signal may vary significantly across iterations.

Credit card fraud detection requires efficient model training while maintaining high accuracy in identifying rare fraudulent transactions. Traditional GBDTs have proven effective in handling imbalanced datasets, but for scenarios involving credit card fraud detection η usually remains constant or follows a predetermined decay schedule throughout the training process. This research introduces Momentum Approximation XGBoost (MAXGBoost), a novel adaptive learning rate approach for GBDTs that dynamically adjusts the learning rate based on loss momentum, enabling faster convergence speed without sacrificing the model's ability to identify rare fraud cases. Unlike traditional GBDTs that require extensive hyperparameter tuning to balance convergence speed with model stability, MAXGBoost automatically adapts η based on the loss landscape, reducing the need for manual optimization while maintaining high detection accuracy (Bushaev, 2017).

3. Related work

XGBoost (Chen & Guestrin, 2016) revolutionized GBDTs by optimizing tree construction and regularization techniques, significantly improving computational efficiency. In credit card fraud detection, Mohan's implementation of XGBoost performed well in the IEEE-CIS Fraud Detection competition (Mohan, 2021). However, the solution relied on static η and therefore suffered from the aforementioned limitations.

Momentum-based η adaptations have been explored in gradient descent methods for neural networks but have seen limited applications in tree-based models. Delta-Bar-Delta Boosting (Wang et al., 2024) introduced an adaptive learning mechanism inspired by momentum-based optimizations. However, it still required extensive hyperparameter tuning and exhibited slow convergence on sparse datasets, making it less practical for fraud detection scenarios.

The AdaBoost framework shows limitations in both speed and accuracy compared to modern XGBoost when handling large-scale datasets. Our work builds upon its adaptive weighting concept but applies it to learning rate adjustments rather than sample weights (Beja-Battais, 2023).

LightGBM enhanced GBDT performance through gradient-based one-side sampling and exclusive feature bundling, significantly reducing memory usage and training time. However, it maintained the same static η limitations as traditional GBDTs, leaving room for improvement in adaptive rate strategies (Ke et al., 2017).

4. Dataset and Features

For our project, we worked with the Credit Card Fraud Detection Dataset (Credit Card Fraud Detection, n.d.). It is a highly imbalanced dataset intended to detect credit card fraud based on the following features:

- **Time**: Time at which the credit card transaction was made.
- **Amount**: Amount of money from the Credit Card transaction.
- **V1-V38**: Attributes V1-V38 are a result of PCA analysis done beforehand.

In our complete dataset, there are a total of 284807 instances. Of these instances, just 492 of them have been marked as credit card fraud. This means that < 0.2% of our data marks the positive class. This imbalance has two major implications: first, models tend to bias heavily towards predicting the majority class, and second, traditional accuracy becomes inadequate for evaluation as a model can achieve high accuracy by always predicting the majority class.

We split our dataset into the following categories:

- **Training**: This is used for training our models on the dataset. This comprises 68% of our data.
- **Validation**: This subset is used for hyperparameter tuning in our model. Validation data comprises 12% of our dataset.
- **Testing**: This subset is used to analyze final performance of the model, as a measure of how well they will work in the real world. The remaining 20% of our dataset is composed of this data.

The dataset comprises real-world e-commerce transactions, each characterized by numerous features including temporal information, card identifiers, merchant data, and transaction amounts. What makes this dataset particularly challenging is its high-dimensional nature combined with extreme sparsity. Many features, such as merchant identifiers and card IDs, exhibit high cardinality, meaning they can take on many possible values, while individual merchants or cards appear very rarely in the dataset. This sparsity creates significant challenges for machine learning approaches that rely on dense feature representations.

5. Methods

Traditionally, neural network weights are updated through the following method,

$$w_{t+1} = w_t - \eta \nabla L \tag{1}$$

where w_{t+1} is the weight of the next iteration, w_t is the current weight, η is the learning rate, and ∇L is the gradient of the loss function. By applying momentum (Bushaev, 2017), gradient descent to a minimum can be made faster through the following modifications given in equations (2) and (3),

$$v_{t+1} = \beta v_t + (1 - \beta)\nabla L \tag{2}$$

$$w_{t+1} = w_t - \eta v_{t+1} \tag{3}$$

where β controls how influential past velocities are on the new velocity calculated, and v is the velocity. In our approach, we look towards an approximation heuristic that builds off of this concept, summarized in equations (4) and (5).

$$v_{t+1} = \beta v_t + (1 - \beta)(L_{t+1} - L_t) \tag{4}$$

$$\eta_{t+1} = \eta_t(1 + v_{t+1}) \tag{5}$$

In this approach, instead of modifying the gradient function as a whole, we specifically look towards tuning η through the momentum approach, whereas η was previously constant in the neural network example. This new value of η_{t+1} is then plugged into equation (1) for gradient boosting.

First, we approximate ∇L as the rate of change between the loss of the current and previous iterations of the GBDT. We then update η specifically based on the velocity. If the loss increases, η increases as it searches for a different local minimum. If the loss decreases, η decreases, attempting to converge to the local minimum it is at. By doing this approximation, we are still able to use a momentum-based heuristic approach within the fast and robust XGBoost framework, a proven framework for winning Kaggle competitions.

6. Experiments

The Decision Tree (DT) and Random Forest (RF) were imported from the sklearn library (Pedregosa et al., 2018). The GBDT models included Constant η model (CGB), Exponential decay model (EGB; Learning Rate, n.d.), and MAXGBoost. We selected the following hyperparameters:

- DT: Max depth = 4. This depth resulted in the best accuracy out of all of the possible depths.
- RF: Estimators = 5. This value resulted in the best accuracy on the range from 1 to 128.
- All XGBoost Models: 423 estimators (iterations of gradient boosting). This value resulted in the best accuracy on n ∈ (1, 500).
    - CGB: $\eta = 0.08$. This value resulted in the best accuracy on $\eta \in (0.01, 1)$
    - EGB: Multiply by 0.9 every iteration, Initial $\eta = 0.1$. This exponent value gave us our best accuracy on $\eta \in (0.01, 1)$
    - MAXGBoost: Initial $\eta = 0.89$, $\beta = 0.99$. These gave us our best accuracy on $\eta, \beta \in (0.01, 1)$

After testing XGBoost, we compared it to the various different models on the basis of four different metrics: accuracy, precision, recall, and area under the curve (AUC) across 5-fold cross-validation. The best-performing fold for each model has been taken to compare against one another.

7. Results & Discussion

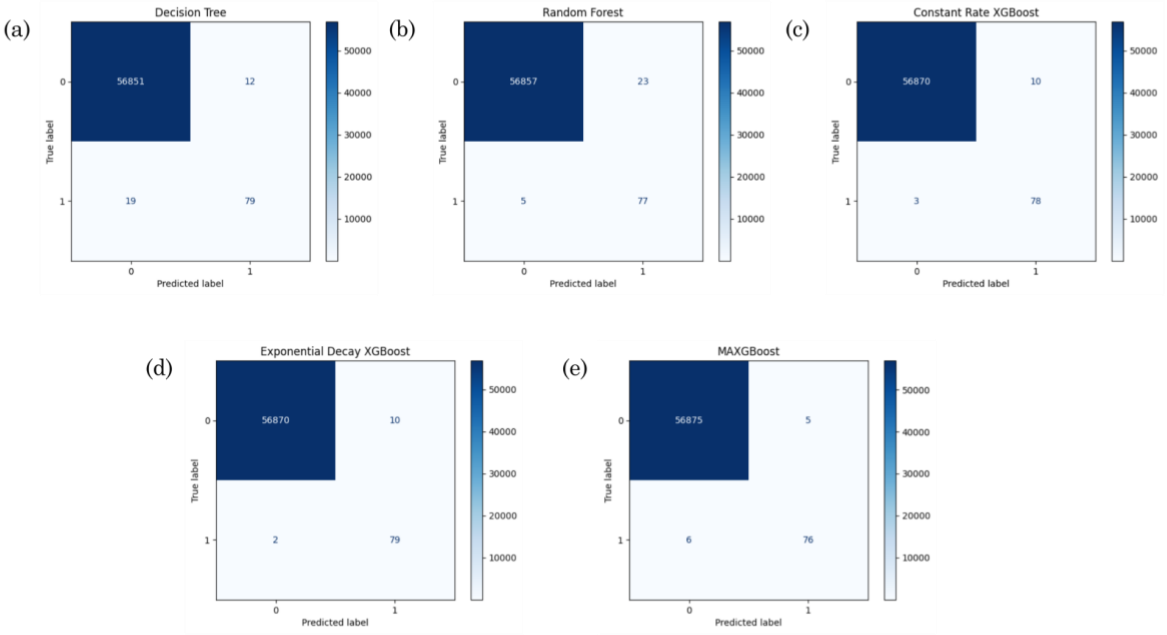
Table 1 presents the performance of the five different models tested on the Credit Card Fraud Detection dataset using the testing set, with confusion matrices provided in Figure 1. The top performing models in this test in terms of pure accuracy are the MAXGBoost, EGB, and CGB models, obtaining scores of 0.99980, 0.99979, and 0.99977 respectively. MAXGBoost also performed best in precision by far with a score of 0.93827, followed by the EGB and CGB models with scores of 0.88764 and 0.88636 respectively. Comparing both the recall and AUC metrics, the EGB model performs best, followed by CGB, RF, MAXGBoost, and then the DT model.

A recurring theme in comparing these models is that they perform exceptionally well in accuracy, but not nearly as well in other metrics. Since > 99% of instances are labeled as negative, the > 99% accuracy is misleading. If this dataset were more balanced, we would expect lower accuracy, more so reflecting the AUC score. As such, we can think of AUC as an alternative to accuracy.

Table 1. Comparison of five different machine learning models in testing accuracy, precision, recall, and AUC.

Model	Accuracy	Precision	Recall	AUC
DT	0.99946	0.86813	0.80612	0.90296
RF	0.99951	0.77000	0.93902	0.96931
CGB	0.99977	0.88636	0.96296	0.98139
EGB	0.99979	0.88764	**0.97531**	**0.98757**
MAXGBoost	**0.99980**	**0.93827**	0.92683	0.96337

Figure 1. Confusion matrices for respective ML models tested.



To mitigate the overfitting tendency of our models, we employed 5-fold cross-validation and a validation set in our dataset split to ensure our models are generalizable.

MAXGBoost’s superior precision indicates it is particularly effective in an environment where false positives can lead to high costs. In this context, where each fraud alert creates both customer friction and operational overhead, MAXGBoost’s ability to minimize false positives can help avoid unnecessary costs. However, MAXGBoost’s slightly lower AUC than RF, CGB, and EGB is an indicator that it wasn’t as effective at identifying the positive class. This could be attributed to MAXBoost's momentum mechanism causing η to decrease too rapidly when it encounters a sequence of small loss improvements. This premature η reduction can prevent the model from exploring regions of the feature space where fraud patterns are more subtle in the dataset. Moreover, the rapid learning rate adaptations can cause MAXGBoost to focus intensely on clearly discriminative features. While this improves precision in the scenario of credit card fraud detection, it may cause the model to underweight features that are weakly correlated with fraud, the minority class. In contrast, RF's random feature sampling, CGB's continual moderate learning rate, and EGB's slower learning rate decay allow them to maintain a more comprehensive coverage of the entire feature space.

8. Conclusions & Future Work

This study demonstrates the effectiveness of decision tree-based models in credit card fraud detection, effectively identifying fraudulent transactions that compromise < 0.2% of the dataset. The EGB and MAXGBoost models with adaptive learning rates especially stood out in this regard, producing accuracy metrics of > 0.999, precision metrics of > 0.88, and recall metrics of > 0.92. Furthermore, the AUC metrics of > 0.96 respectively demonstrate the robustness of these models. EGB excelled in terms of recall and AUC, demonstrating that it is effective in picking out the fraudulent transactions. MAXGBoost produced the best accuracy and precision scores of 0.99938 and 0.93827 respectively, suggesting that it is the most effective model for customer satisfaction and avoiding falsely detecting fraudulent transactions made by the real customer.

Looking ahead, there are several promising directions for future research. Given the strong performance of both EGB and MAXGBoost in different areas, a natural next step would be developing a hybrid momentum-decay approach. This could be further built upon through feature-specific momentum calculations or class-aware momentum adjustments. Additionally, developing techniques to handle concept drift would be beneficial, as fraudsters evolve their strategies overtime.

9. Reproducibility

All code used for the development and tuning of our machine learning models can be found in the following repository: https://github.com/Vini-41/MAXGBoost.

Code was written in Python 3.12.2.

10. References

Andrej Baranovskij. (2019, March 12). *Selecting Optimal Parameters for XGBoost Model Training*. Medium; Towards Data Science. https://medium.com/towards-data-science/selecting-optimal-parameters-for-xgboost-model-training-c7cd9ed5e45e

Beja-Battais, P. (2023, October 6). *Overview of AdaBoost: Reconciling its views to better understand its dynamics*. ArXiv.org. https://doi.org/10.48550/arXiv.2310.18323

Brownlee, J. (2016, September 15). *Tune Learning Rate for Gradient Boosting with XGBoost in Python*. Machine Learning Mastery. https://machinelearningmastery.com/tune-learning-rate-for-gradient-boosting-with-xgboost-in-python/

Chen, T., & Guestrin, C. (2016). XGBoost: a Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*, 785–794. https://doi.org/10.1145/2939672.2939785

Credit Card Fraud Detection. (n.d.). Kaggle. https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud/data

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). *LightGBM: A Highly Efficient Gradient Boosting Decision Tree*. Neural Information Processing Systems; Curran Associates, Inc. https://papers.nips.cc/paper_files/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html

Learning rate. (n.d.). Wikipedia. https://en.wikipedia.org/wiki/Learning_rate

Mohan, A. (2021, April 6). *IEEE-CIS Fraud Detection - Top 5% Solution - Towards Data Science*. Medium; Towards Data Science. https://medium.com/towards-data-science/ieee-cis-fraud-detection-top-5-solution-5488fc66e95f

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Müller, A., Nothman, J., Louppe, G., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2018). Scikit-learn: Machine Learning in Python. *ArXiv:1201.0490 [Cs]*. https://arxiv.org/abs/1201.0490

Vitaly Bushaev. (2017, December 4). *Stochastic Gradient Descent with momentum - Towards Data Science*. Medium; Towards Data Science. https://medium.com/towards-data-science/stochastic-gradient-descent-with-momentum-a84097641a5d

Wang, C., Wang, Z., Ouyang, Y., & Soleimani, B. H. (2024, May 27). *Adaptive Learning Rates for Gradient Boosting Machines*. Proceedings of the Canadian Conference on Artificial Intelligence. https://caiac.pubpub.org/pub/py65wd3c/release/1