XGBoost
Epoch IIT Hyderabad

Chakka Surya Saketh
AI22BTECH11005

Introduction

XGBoost is an ensemble learning method. Sometimes, it may not be sufficient to rely upon the results of just one machine learning model. Ensemble learning offers a systematic solution to combine the predictive power of multiple learners. The beauty of this powerful algorithm lies in its scalability, which drives fast learning through parallel and distributed computing and offers efficient memory usage. In short for Extreme Gradient Boosting, is a machine learning algorithm that has demonstrated outstanding performance across various domains, ranging from finance and healthcare to natural language processing and computer vision. The accuracy it consistently gives, and the time it saves, demonstrates how useful it is.

Working Principles

The core concept of XGBoost lies in the sequential training of models, where each new model is constructed to correct the errors made by the previous ones. This is achieved through gradient boosting, a technique that minimizes the loss function by iteratively adding models.

The distinguishing features of XGBoost include its regularized boosting framework, which prevents overfitting by incorporating regularization terms into the loss function. Additionally, XGBoost employs a novel approach called ”tree pruning” to minimize the depth of trees, thereby reducing their complexity and enhancing generalization.

Unique Features

- Regularization: XGBoost has an option to penalize complex models through both L1 and L2 regularization. Regularization helps in preventing overfitting.
- Handling sparse data: Missing values or data processing steps like one-hot encoding make data sparse. XGBoost incorporates a sparsity-aware split finding algorithm to handle different types of sparsity patterns in the data.
- Weighted quantile sketch: Most existing tree based algorithms can find the split points when the data points are of equal weights (using quantile sketch algorithm). However, they are not equipped to handle weighted data. XGBoost has a distributed weighted quantile sketch algorithm to effectively handle weighted data.
- Block structure for parallel learning: For faster computing, XGBoost can make use of multiple cores on the CPU. This is possible because of a block structure in its system design. Data is sorted and stored in in-memory units called blocks. Unlike other algorithms, this enables the data layout to be reused by subsequent iterations, instead of computing it again. This feature also serves useful for steps like split finding and column sub-sampling.
- Cache awareness: In XGBoost, non-continuous memory access is required to get the gradient statistics by row index. Hence, XGBoost has been designed to make optimal use of hardware. This is done by allocating internal buffers in each thread, where the gradient statistics can be stored.
- Out-of-core computing: This feature optimizes the available disk space and maximizes its usage when handling huge datasets that do not fit into memory.

Hyper parameters

- max_depth: The depth of the individual decision tree.
- gamma: The minimum loss reduction is required to make a further partition on a leaf node of the tree.
- lambda: L2 regularisation term.
- alpha: L1 regularisation term on weights.
- num_class: The number of classes to classify the data into. This should be set for multi-class problems. default is binary.
- n_jobs: Number of CPU core to use during training.

Conclusion

XGBoost, with its ability to handle complex relationships, regularization techniques, and efficient boosting framework, stands out as a powerful machine learning algorithm. Its versatility, performance, and interpretability have contributed to its widespread adoption in various industries and research domains. While XGBoost comes with certain limitations, its strengths far outweigh its drawbacks, making it a valuable tool for practitioners seeking high-performance predictive modeling.