Tree boosting is an effective and widely used machine learning approach. In this work, we present XGBoost, a scalable end-to-end tree boosting system widely used by data scientists to produce cutting-edge results on a wide range of machine learning problems. The most important factor in XGBoost's success is its capacity to scale in all settings. The scalability of XGBoost is due to a number of significant system and algorithmic improvements.

The following are the paper's key contributions:

- We design and build a highly scalable end-to-end tree boosting solution.
- We provide a theoretically motivated weighted quantile sketch for efficient proposal calculation.
- We offer a one-of-a-kind sparsity-aware parallel tree learning approach.
- We provide an efficient cache-aware block structure for out-of-core tree learning.

Basic Exact Greedy Algorithm

One of the most challenging problems in tree learning is determining the best split. A split discovery algorithm iterates over all of the potential splits on all of the attributes to accomplish this. This is referred to as the precise greedy algorithm. The exact greedy approach is supported by the majority of existing single machine tree boosting implementations, including scikit-learn, R's gbm, and the single machine version of XGBoost.

Approximate Algorithm

Our system efficiently supports both exact greedy and approximation algorithms in all conditions, including both local and global proposal techniques. Users can easily select one of the methods based on their need.

Cache-aware Access

The exact greedy method can be alleviated by a cache-aware prefetching mechanism. On specifically, in each thread, we allocate an internal buffer, load the gradient statistics into it, and then do mini-batch accumulation. When the number of rows is large, this prefetching changes the direct read/write dependency into a longer dependency, which helps to reduce runtime cost.

In this paper, we highlighted the lessons we acquired while building XGBoost. XGBoost can solve real-world size difficulties with a small amount of resources.