Block-distributed Gradient Boosted Trees

Theodore Vasiloudis[1,3], Hyunsu Cho[2], Henrik Boström[3]

[1] Research Institutes of Sweden, [2] Amazon Web Services, [3] KTH Royal Institute of Technology.

Summary

We introduce block-distributed training for gradient boosted trees (GBT), enhancing their scalability.
Our contributions are the following:

- The first algorithm for data-and-feature parallel training of GBTs.
- We achieve orders of magnitude improved communication cost by taking advantage of data sparsity.

Introduction

Gradient Boosted Trees: One of the most widely used algorithms in IR tasks like learning-to-rank and CTR prediction.

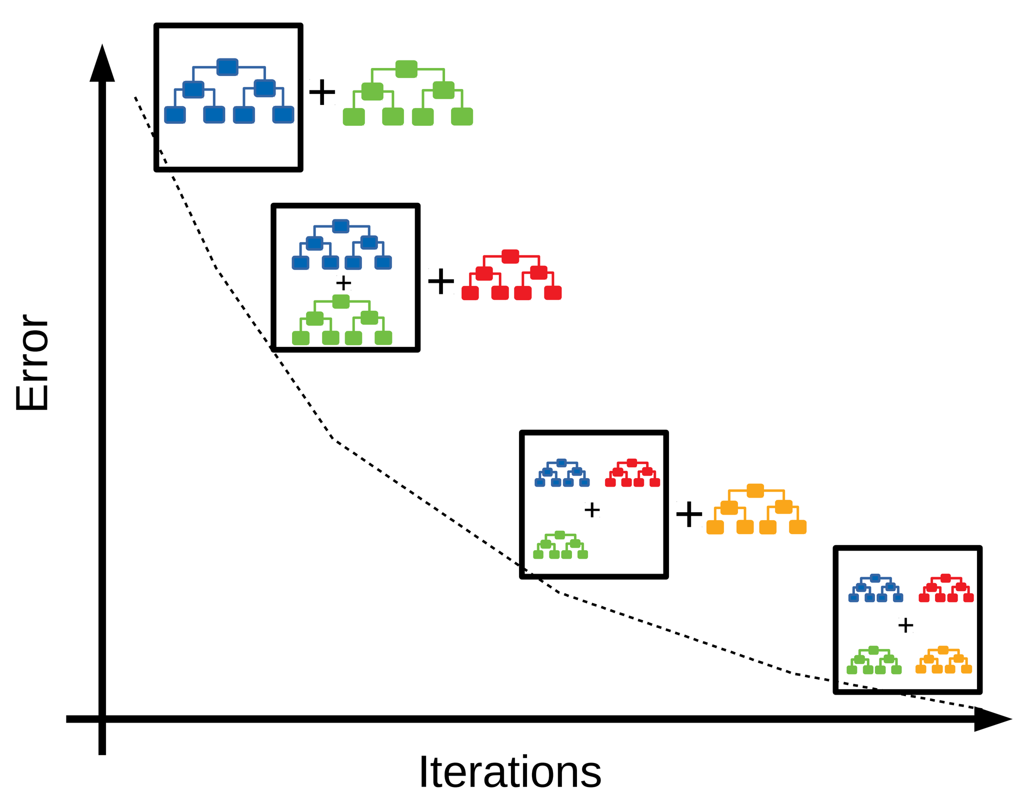
Why it's important:

- We need highly scalable algorithms for enterprise scale, *high-dimensional data*.
- Training on such data is done in clusters, where *network communication* is the main bottleneck.

Why it's difficult:

- All systems *only use row distribution*, and for feature parallel training, assume *all data fit into the memory* of each worker.
- In addition, they use *dense communication*, leading to redundant network traffic.

Gradient Boosted Trees



Block-distributed data

Idx	Feat. 1	Feat. 2
1	13	0
2	7	1

(a) Worker 1.

Idx	Feat. 3	Grad.
1	488	1.5
2	667	2.5

(b) Worker 2.

Idx	Feat. 1	Feat. 2
3	3.5	0
4	1.6	2

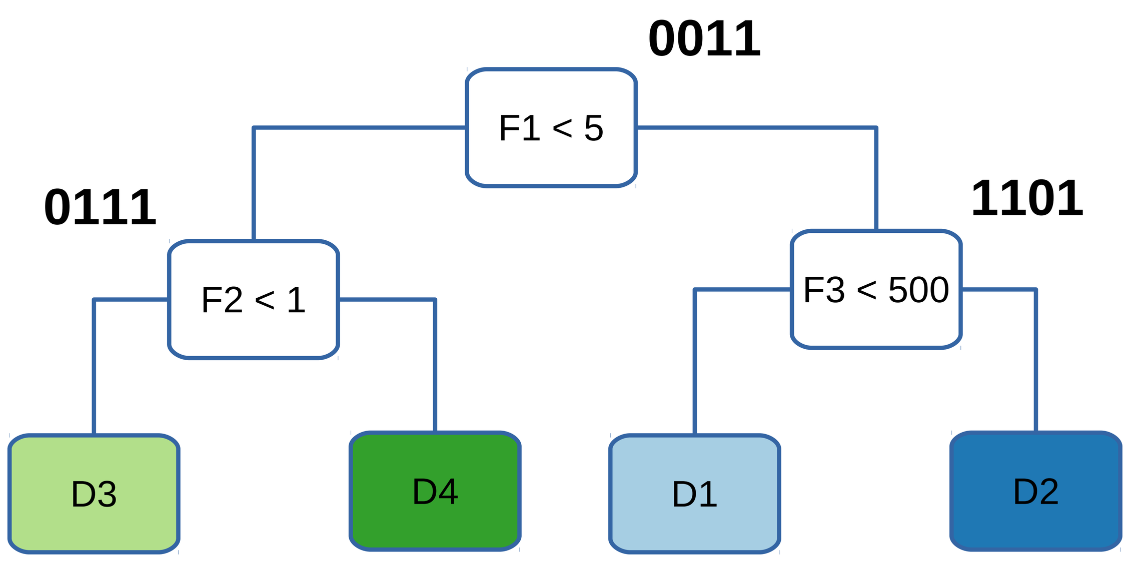
(c) Worker 3.

Idx	Feat. 3	Grad.
3	122	2
4	366	2.5

(d) Worker 4.

Quickscorer

Uses bitstrings to quickly determine exit leaf.



Block-distributed Quickscorer

- Use Quickscorer at each worker locally.
- Communicate bitstrings to get exit leaf.

Idx	Bitstring
1	0011
2	0011

(a) Worker 1.

Idx	Bitstring
1	1111
2	1101

(b) Worker 2.

Idx	Bitstring
3	1111
4	0111

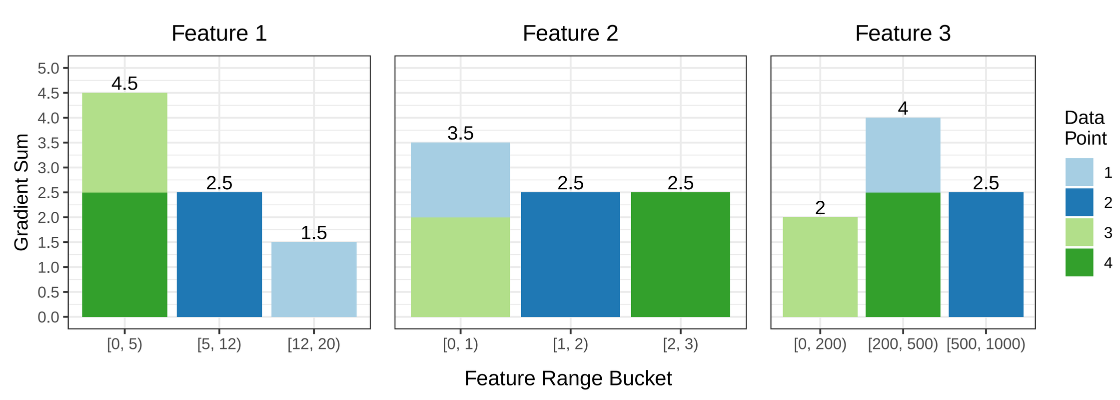
(c) Worker 3.

Idx	Bitstring
3	1111
4	1111

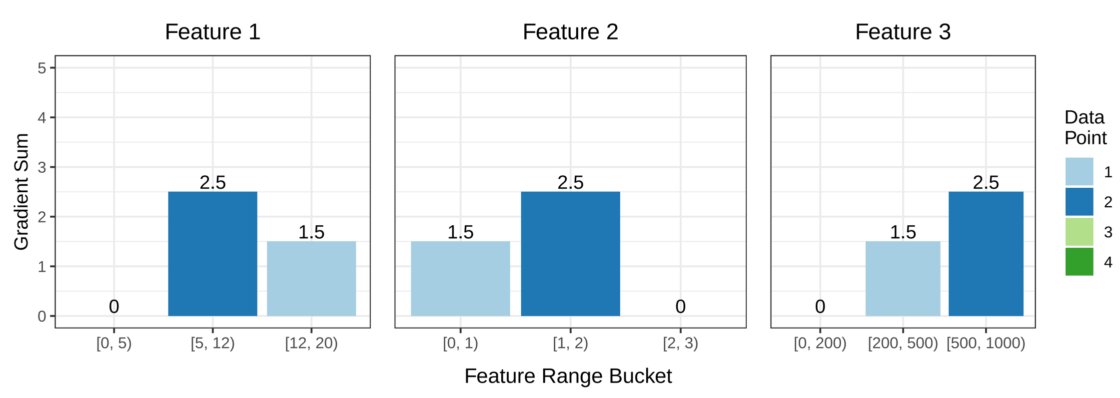
(d) Worker 4.

Gradient Histograms

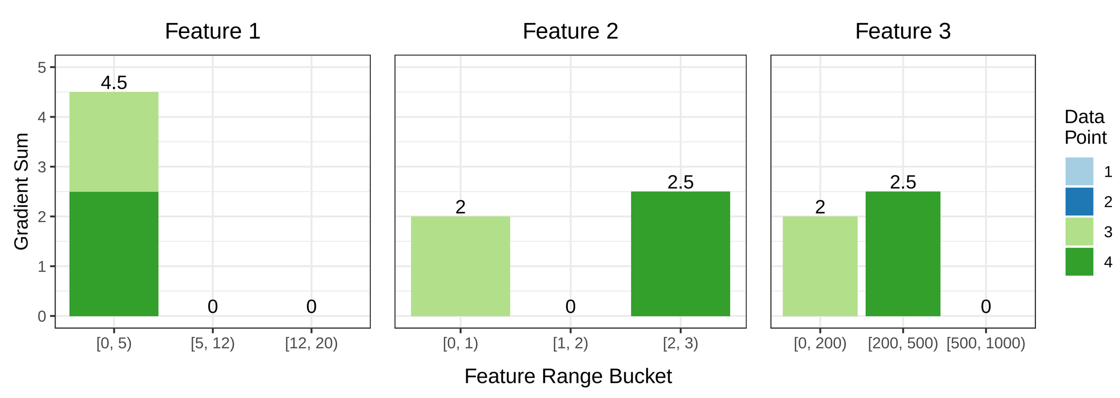
- The most computationally intensive part of GBT training is gradient histogram calculation.
- We use gradient histograms to calculate the potential accuracy gain of splitting a leaf.



Gradient Histograms are Sparse

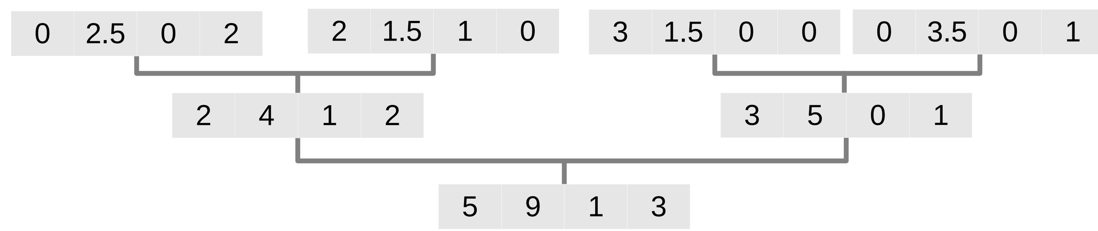


(a) Local gradient histogram for Worker 1.



(b) Local gradient histogram for Worker 2.

Dense Communication



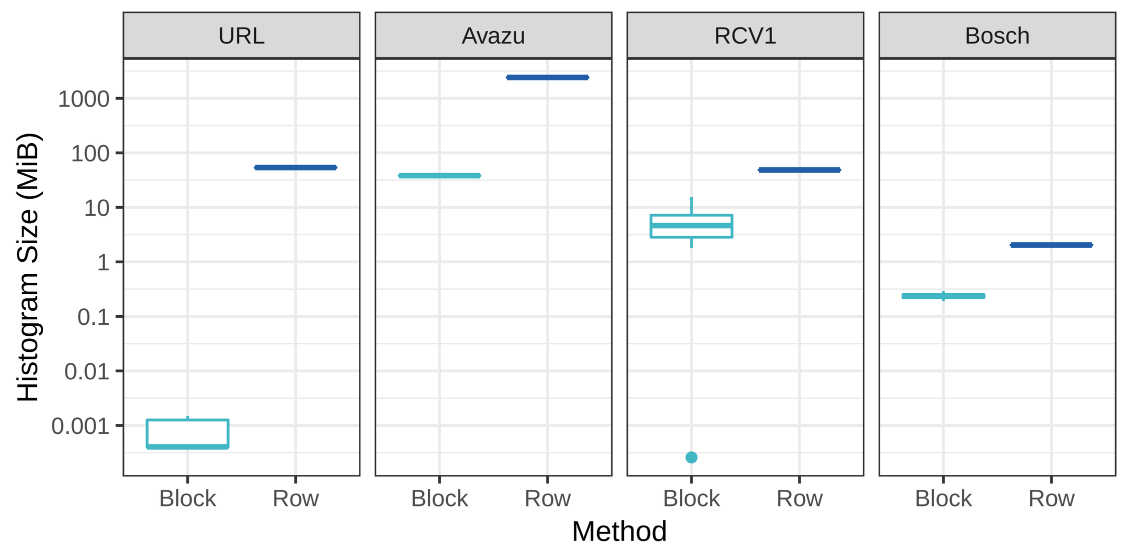
Sparse Communication

We create sparse matrices for the histograms, and communicate those, using the Parameter Server.

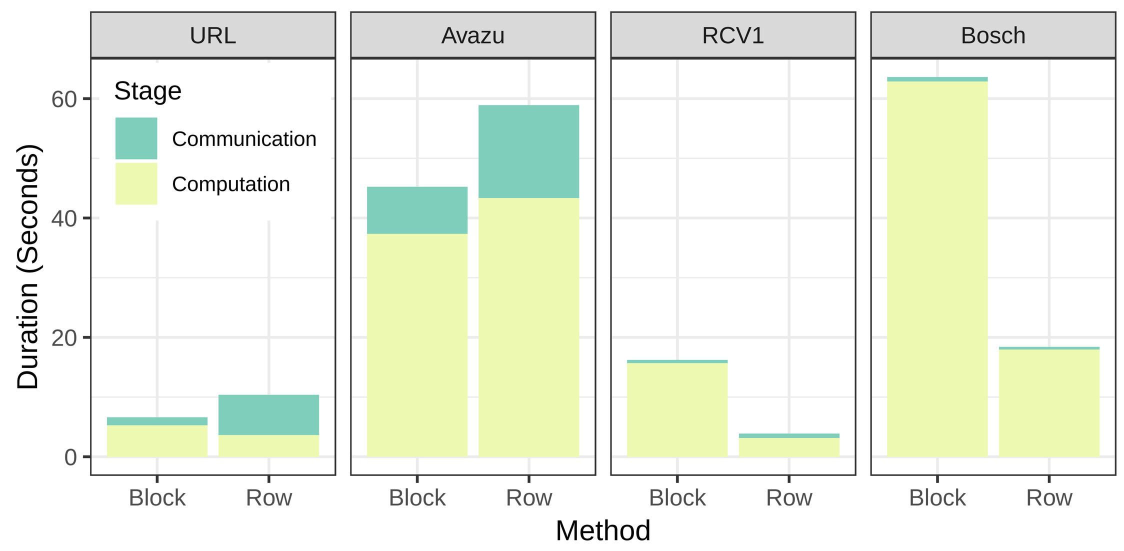
$$\begin{vmatrix} 0 & 0 & 0 & 0 \\ 5 & 8 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 6 & 0 & 0 \end{vmatrix} \equiv \begin{aligned} A &= [5, 8, 3, 6] \\ IA &= [0, 0, 2, 3, 4] \\ JA &= [0, 1, 2, 1] \end{aligned}$$

Using a sparse format we can significantly shrink the number of values being communicated.

Results



(a) Byte size of histograms being communicated for block (light blue) and row (dark blue) distributed approach.



(b) Time for histogram creation, including computation and communication.

Conclusions

- Several orders of magnitude communication savings are possible for highly sparse data.
- More work needed to offset the computational overhead of the sparse data structures.

Contact

- Email: tvas@kth.se
- Twitter: @thvasilo

URL: