XGBoost: A Scalable Tree Boosting System

Presenter: Tianqi Chen

University of Washington

Outline

- Introduction: Trees, the Secret Sauce in Machine Learning
- Parallel Tree Learning Algorithm
- Reliable Distributed Tree Construction

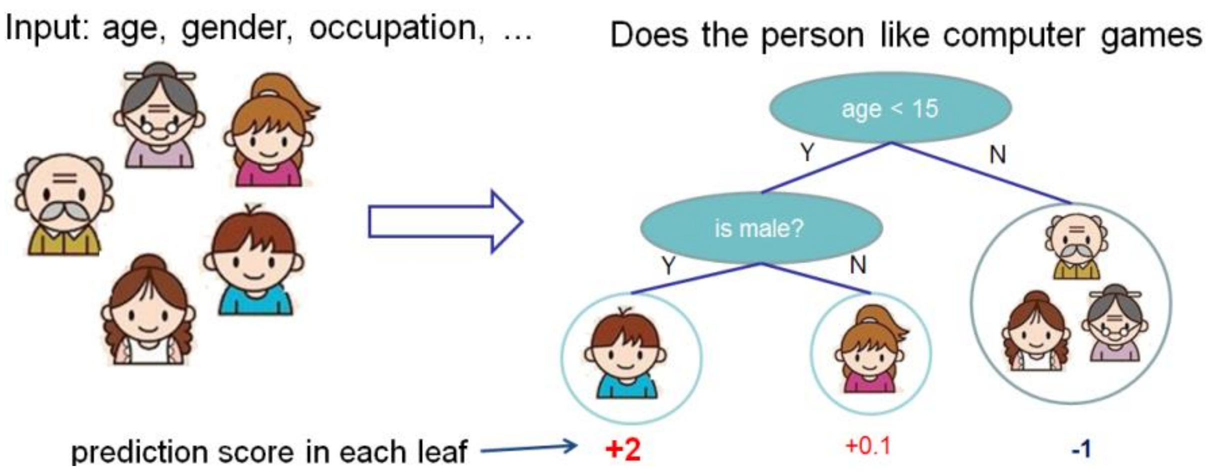
Machine Learning Algorithms and Common Use-cases

- Linear Models for Ads Clickthrough
- Factorization Models for Recommendation
- Deep Neural Nets for Images, Audios etc.

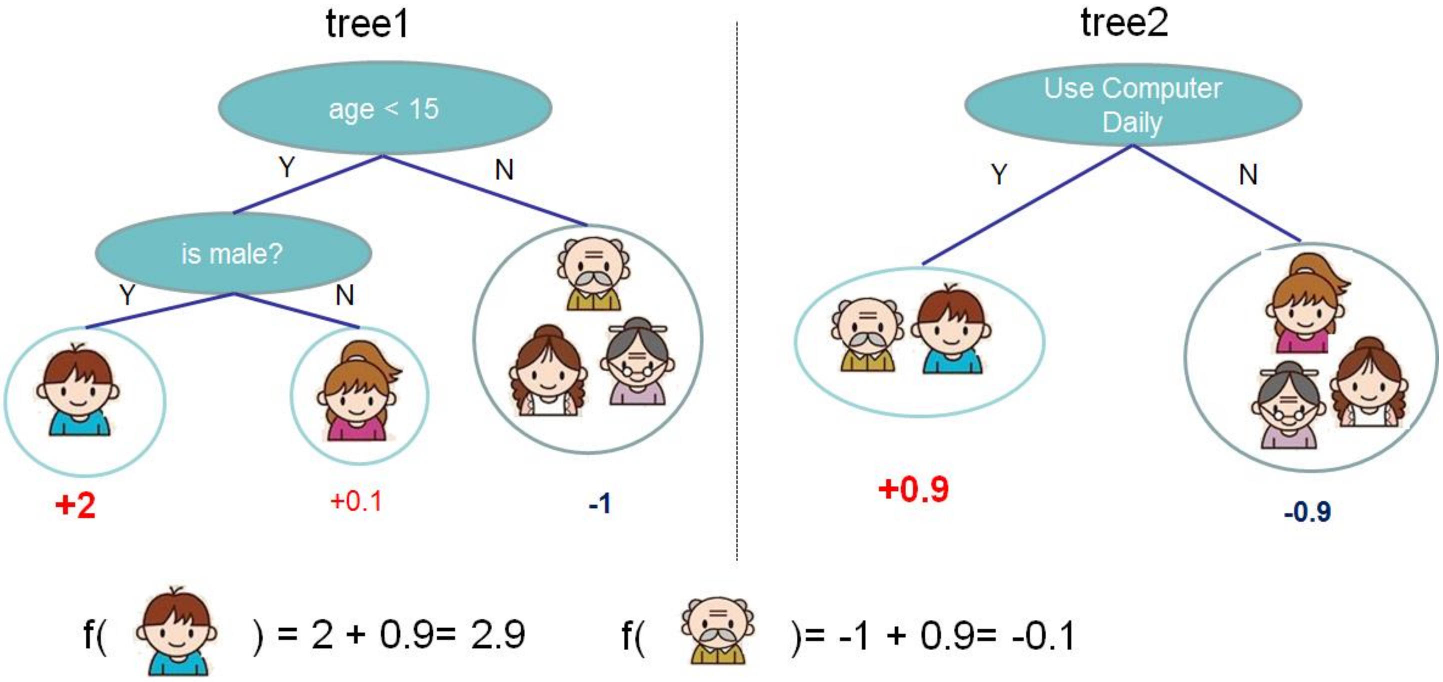
- **Trees** for tabular data with **continuous inputs**: the secret sauce in machine learning
  - Anomaly detection
  - Action detection
  - From sensor array data
  - ....

Regression Tree

- Regression tree (also known as CART)
- This is what it would looks like for a commercial system



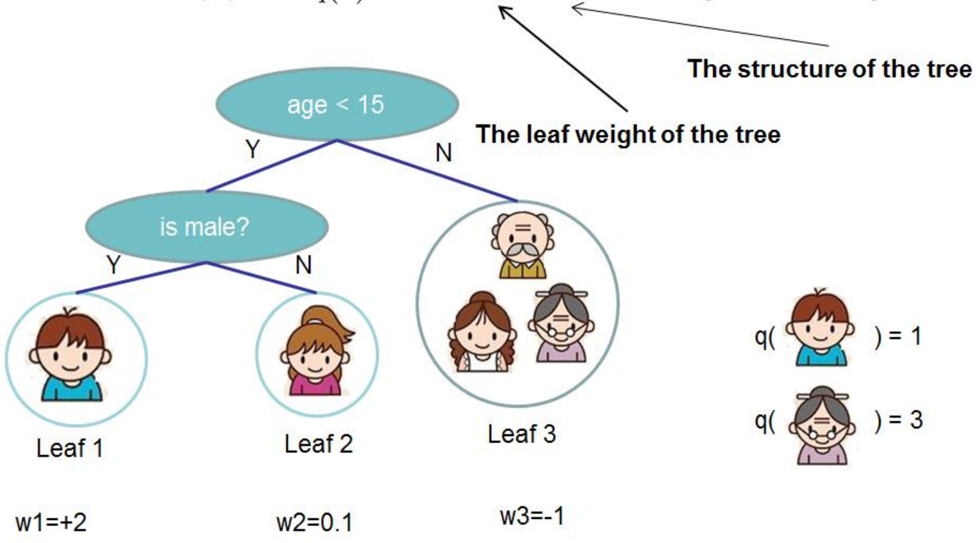
When Trees forms a Forest (Tree Ensembles)



Learning a Tree Ensemble in Three Slides

Model $\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), \quad f_k \in \mathcal{F}$

$$f_t(x) = w_{q(x)}, \quad w \in \mathbf{R}^T, q : \mathbf{R}^d \to \{1, 2, \cdots, T\}$$



Learning a Tree Ensemble in Three Slides

Objective

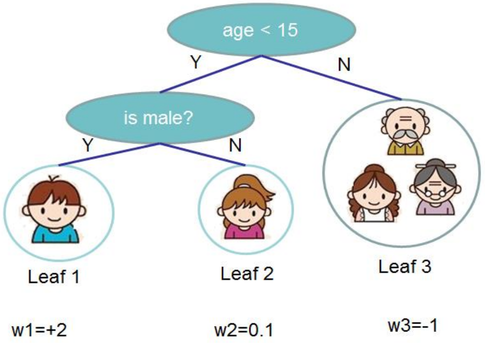
$$Obj = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

Training Loss measures how well model fit on training data

Regularization, measures complexity of trees

$$\Omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$$

Number of leaves **L2 norm of leaf scores**

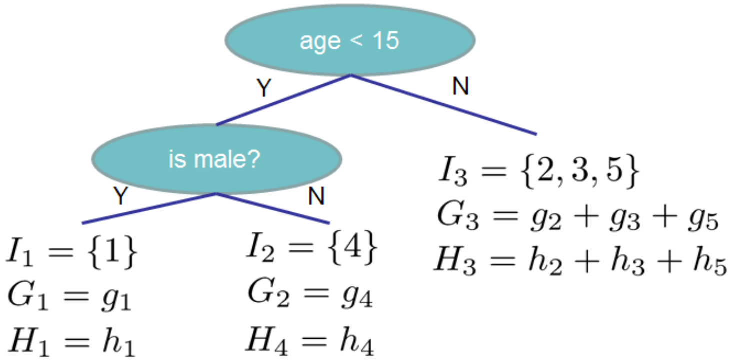


$$\Omega = \gamma 3 + \frac{1}{2}\lambda(4 + 0.01 + 1)$$

Learning a Tree Ensemble in Three Slides

Score for a new tree

Instance index	gradient statistics
1	g1, h1
2	g2, h2
3	g3, h3
4	g4, h4
5	g5, h5



$$Obj = -\sum_j \frac{G_j^2}{H_j + \lambda} + 3\gamma$$

The smaller the score is, the better the structure is

Gradient Statistics

$$g_i = \partial_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)}), \quad h_i = \partial^2_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$$

Outline

- Introduction: Trees, the Secret Sauce in Machine Learning
- Parallel Tree Learning Algorithm
- Reliable Distributed Tree Construction
- Results

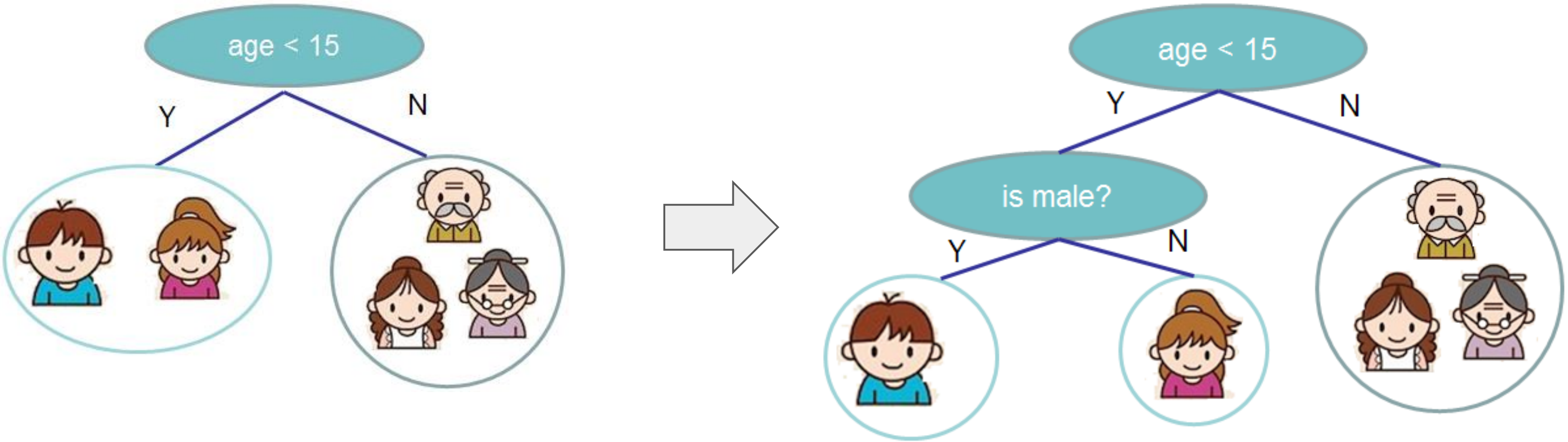
Tree Finding Algorithm

- Enumerate all the possible tree structures
- Calculate the structure score, using the scoring eq.

$$Obj = -\frac{1}{2}\sum_{j=1}^{T}\frac{G_j^2}{H_j+\lambda}+\gamma T$$

- Find the best tree structure
- But… there can be many trees

Greedy Split Finding by Layers

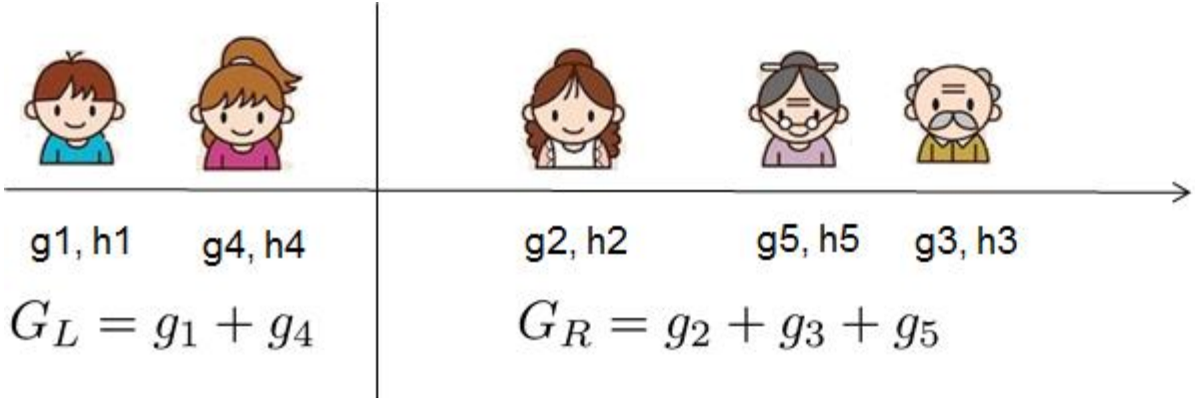


Split Finding Algorithm on Single Node

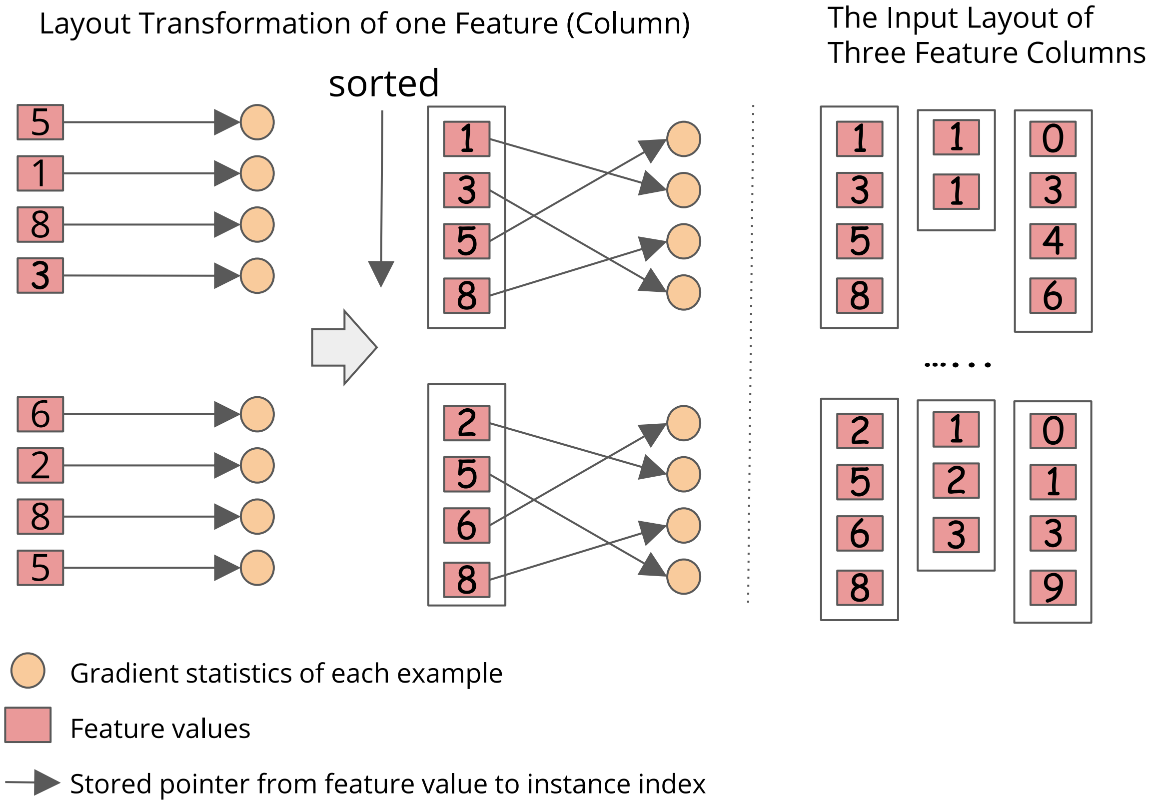
Scan from left to right, in sorted order of feature

Calculate the statistics in one scan

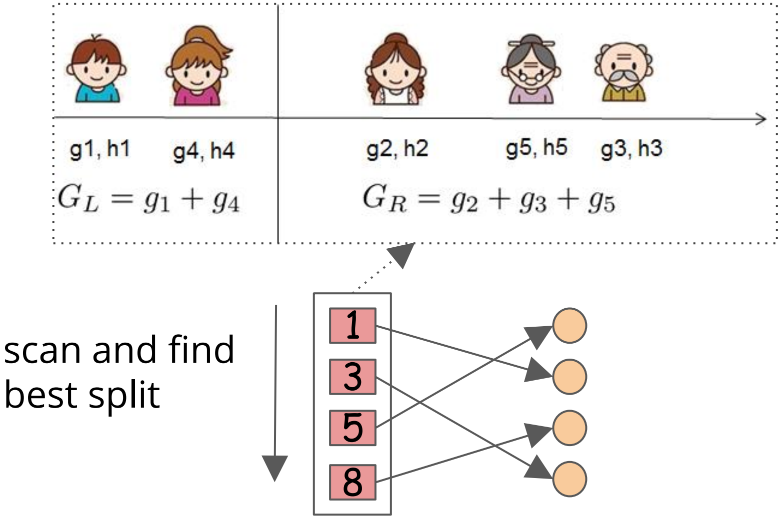
However, this requires **sorting** over features - O(n logn) per tree



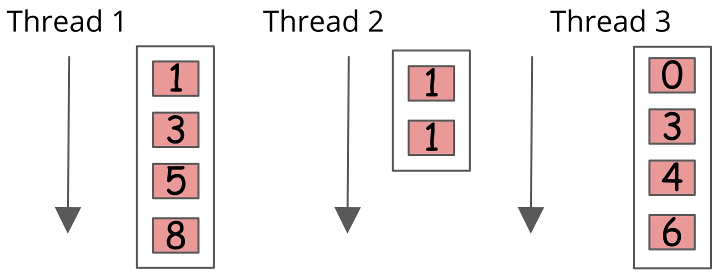
The Column based Input Block



Parallel Split Finding on the Input Layout



Parallel scan and split finding



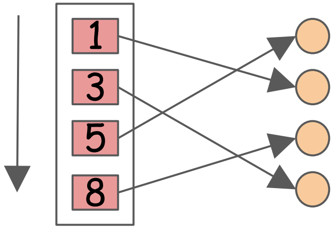
Gradient statistics of each example

Feature values

Stored pointer from feature value to instance index

Cache Miss Problem for Large Data

scan and find
best split



Gradient statistics of each example

Feature values

Stored pointer from feature value to instance index

G = G + g[ptr[i]]
H = H + h[ptr[i]]

calculate score....

G = G + g[ptr[i]]
H = H + h[ptr[i]]

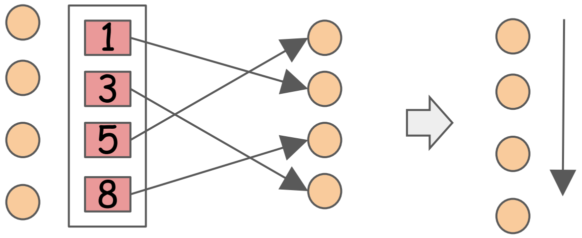
Short range instruction dependency, with **non-contiguous** access to g

Cause **cache miss** when g does not fit into cache

Use **prefetch** to change dependency to long range.

Cache-aware Prefetching

prefetch



scan and find best split

Gradient statistics of each example

Feature values

Stored pointer from feature value to instance index

bufg[1] = g[ptr[1]]
bufg[2] = g[ptr[2]]
...

G = G + bufg[1]

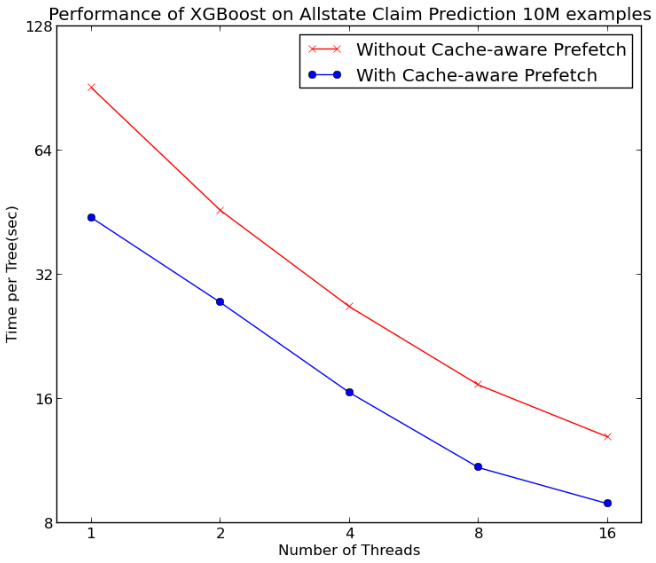
calculate score ...

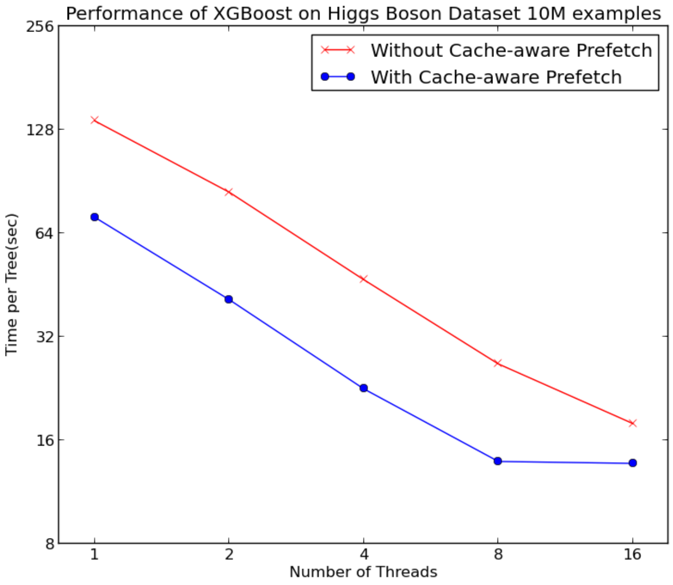
G = G + bufg[2]

Long range instruction dependency

Continuous memory access

Impact of Cache-aware Prefetch (10M examples)



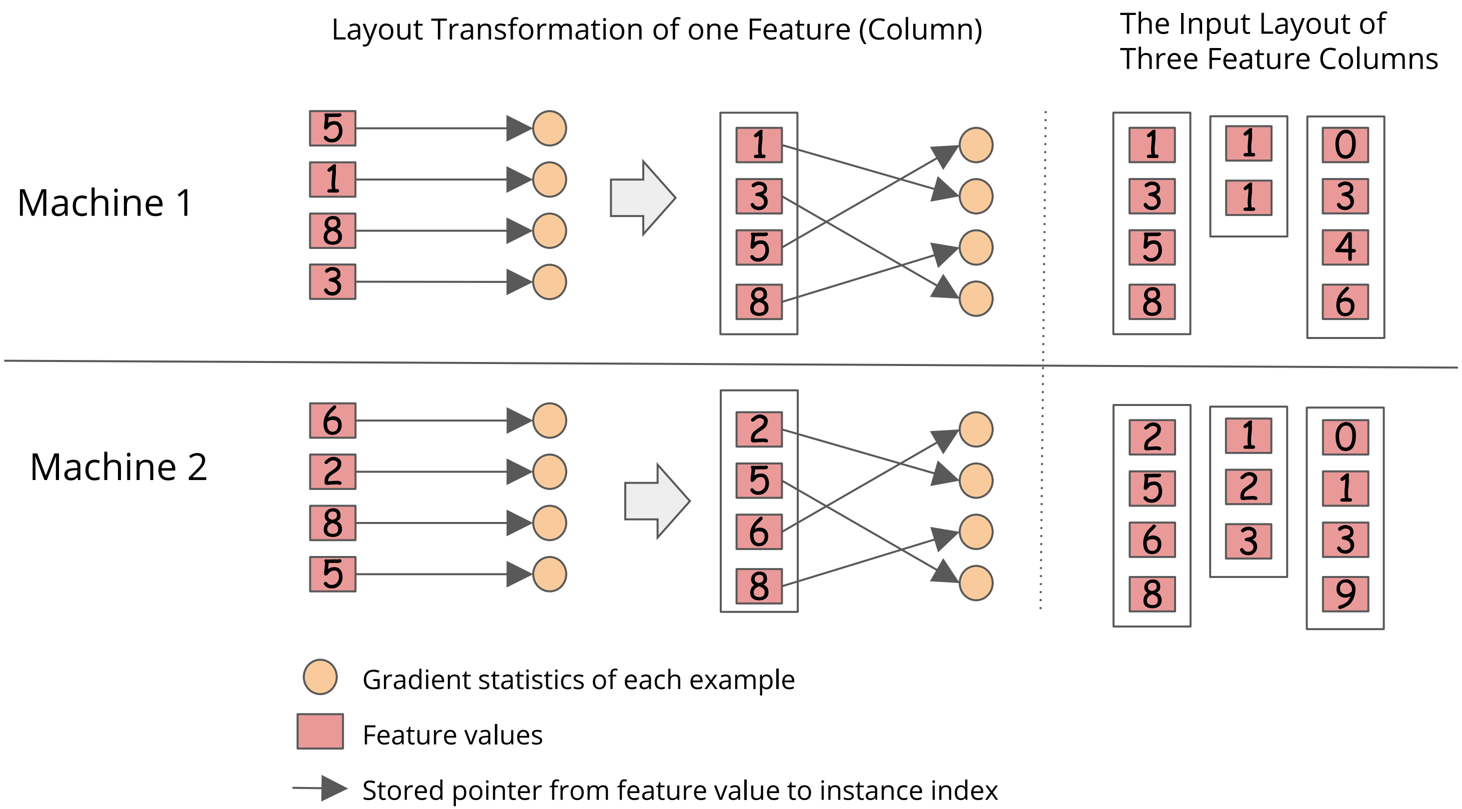


Effect of Cache-miss kicks in,
prefetch makes things ***two times*** faster

Outline

- Introduction: Trees, the Secret Sauce in Machine Learning
- Parallel Tree Learning Algorithm
- Reliable Distributed Tree Construction
- Results

The Distributed Learning with same Layout

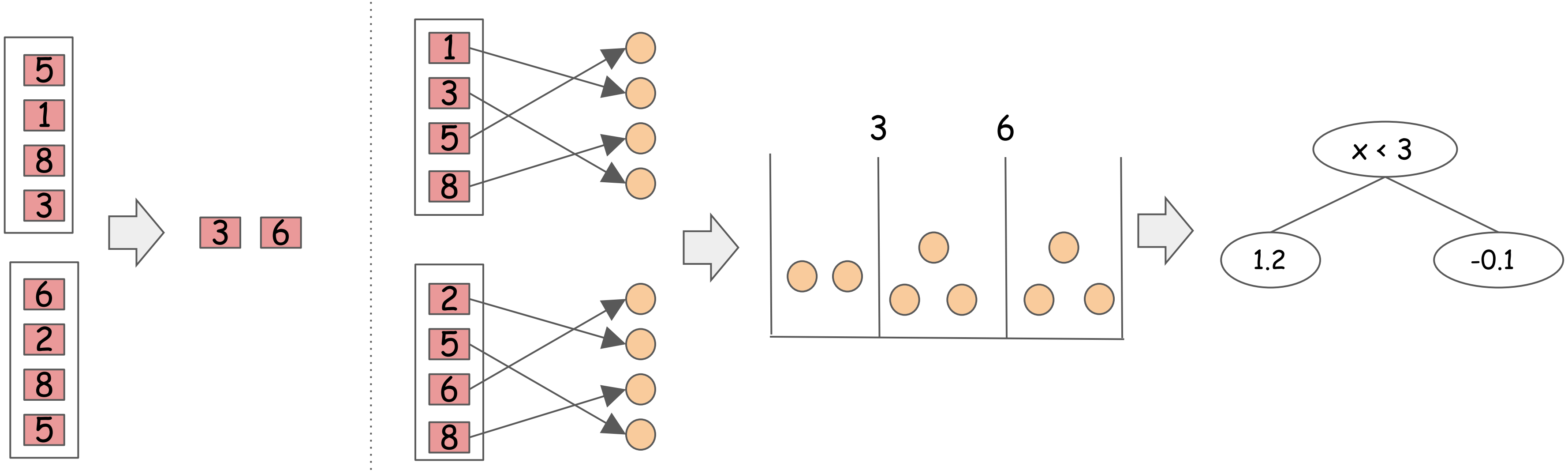


Sketch of Distributed Learning Algorithm

Step 1: Split proposal by **Distributed Weighted Quantile Sketching**

Step 2: **Histogram Calculation**

Step 3: Select Best Split with Structure Score

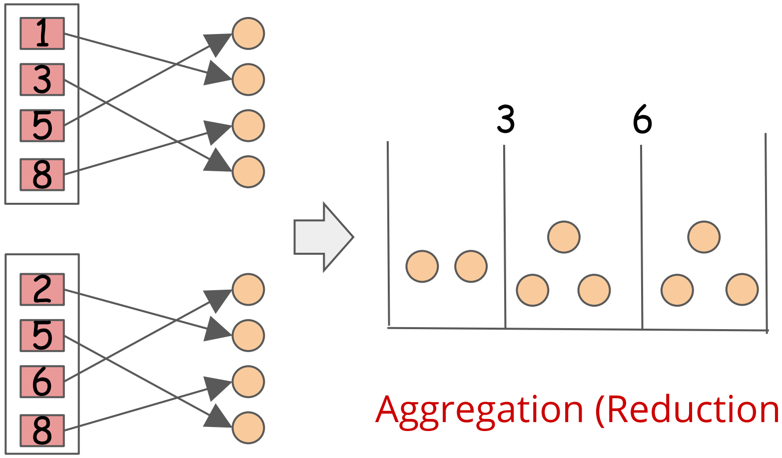


Both steps benefit from Optimized Input Layout!

Why Weighted Quantile Sketch

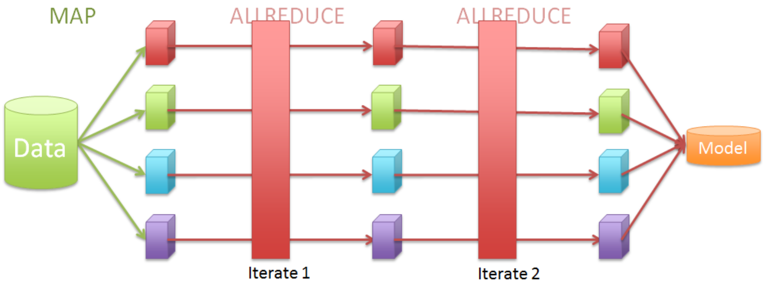
- Enable equivalent proposals among the data
- Data

Communication Problem in Learning



Aggregation (Reduction)

Allreduce

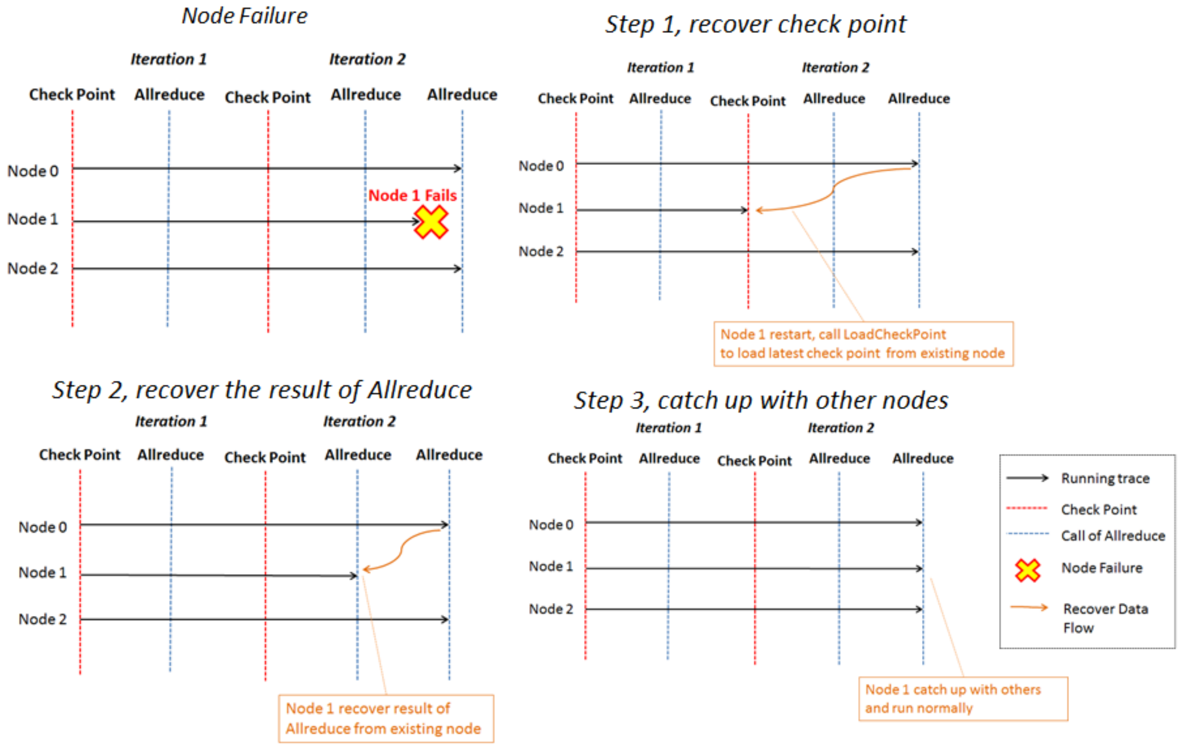


Rabit: Reliable Allreduce and Broadcast Interface

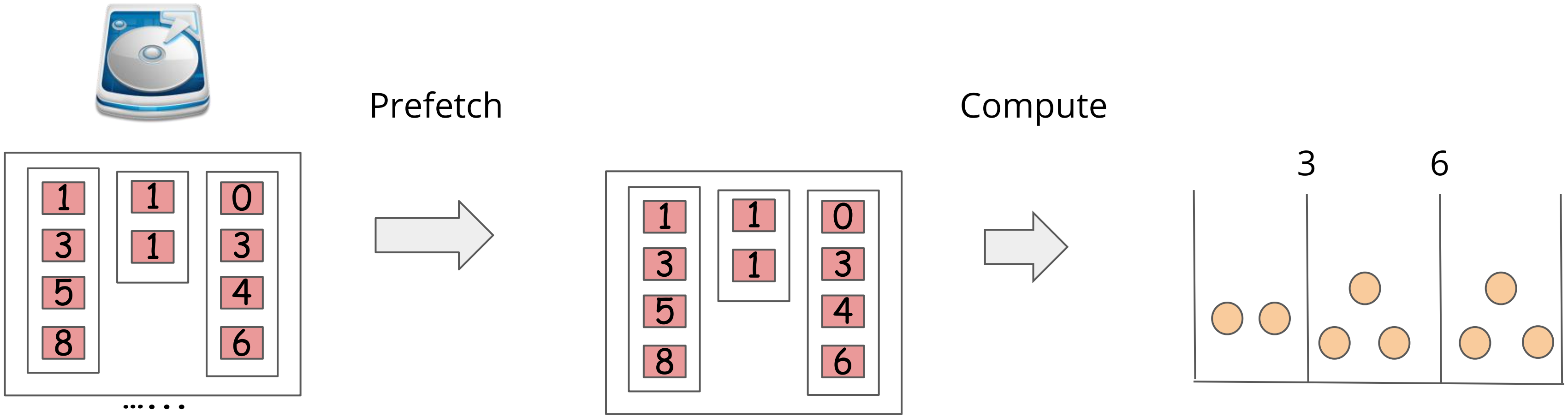
Important Property of Allreduce

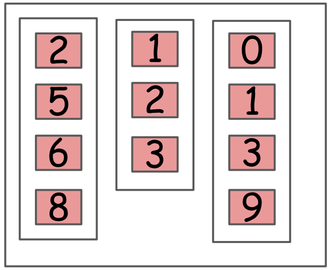
All the machines get the **same** reduction result

Can remember and forward result to failed nodes



Out of Core Version



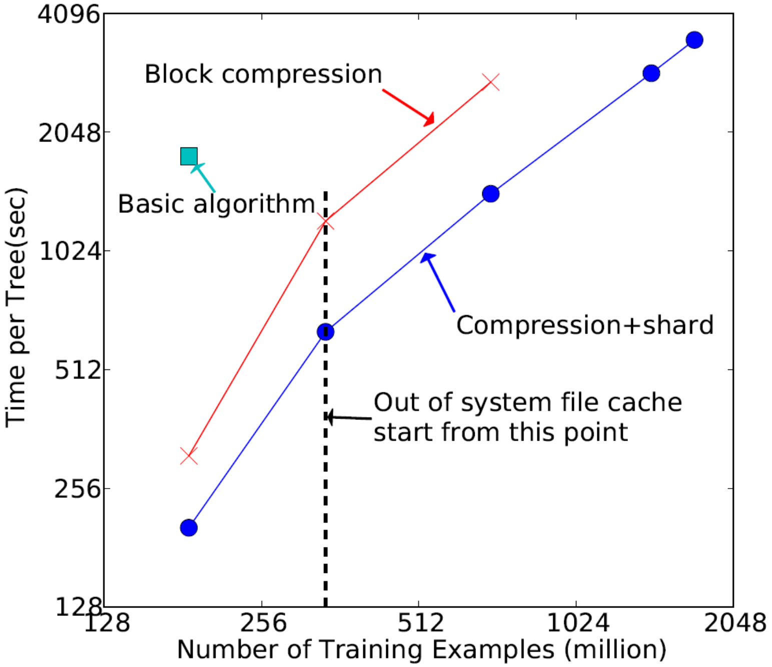


Other optimization techniques

- Block compression
- Disk sharding

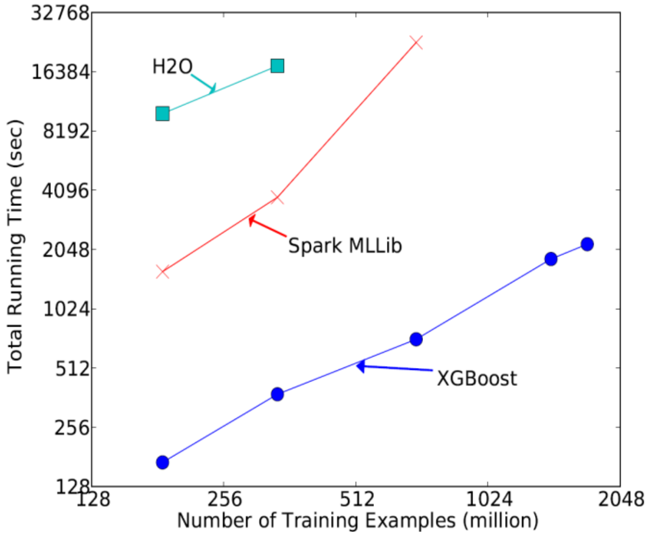
External Memory Version

- Impact of external memory optimizations
- On a single EC2 machine with two SSD

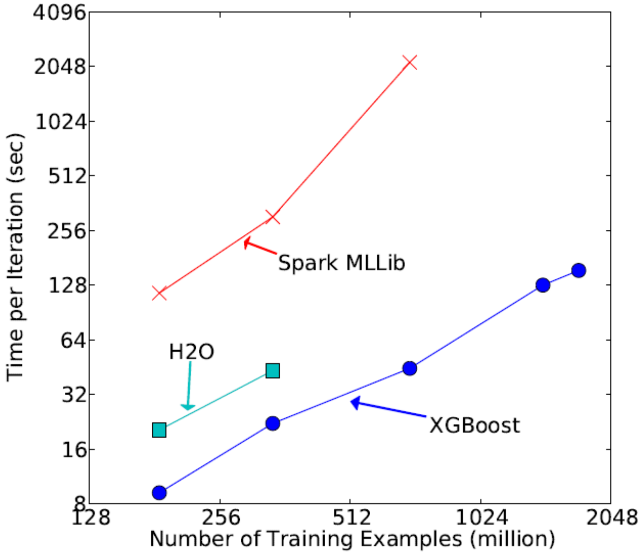


Distributed Version Comparison

Cost include data loading

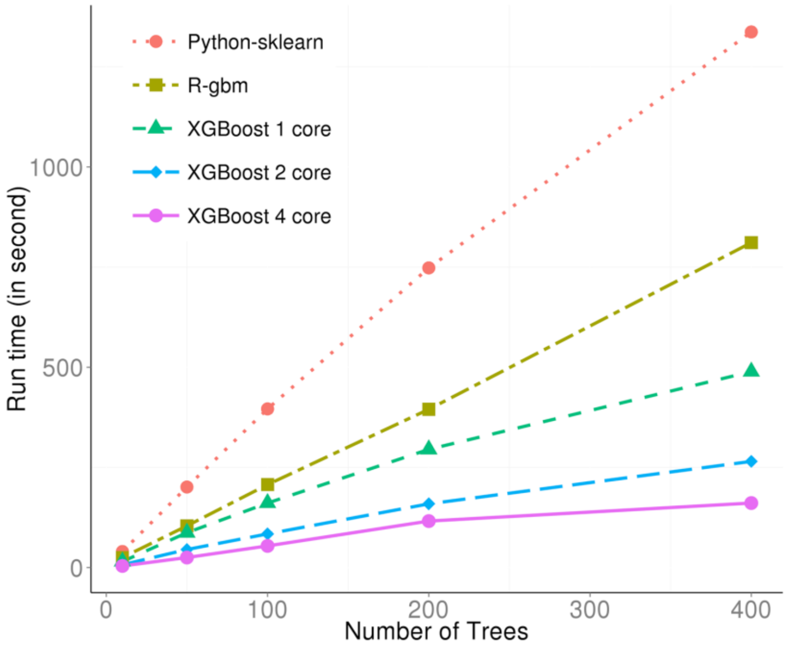


Cost exclude data loading



Comparison to Existing Open Source Packages

Comparison of Parallel XGBoost with commonly used Open-Source implementation of trees on Higgs Boson Challenge Data.



- 2-4 times faster with single core
- Ten times faster with multiple cores

Impact of the System

The most frequently used tool by data science competition winners

17 out of 29 winning solutions in kaggle last year used XGBoost

Solve wide range of problems: store sales prediction; **high energy physics event classification**; web text classification; customer behavior prediction; **motion detection**; ad click through rate prediction; malware classification; product categorization; hazard risk prediction; massive online course dropout rate prediction

Many of the problems used data from sensors

Present and Future of KDDCup. Ron Bekkerman (KDDCup 2015 chair): *"Something dramatic happened in Machine Learning over the past couple of years. It is called XGBoost – a package implementing Gradient Boosted Decision Trees that works wonders in data classification. Apparently, every winning team used XGBoost, mostly in ensembles with other classifiers. Most surprisingly, the winning teams report very minor improvements that ensembles bring over a single well-configured XGBoost.."*

The methods presented in this talk further improves the scale and reliability

Thank You