XGBOOST: A SCALABLE TREE BOOSTING SYSTEM

(T. CHEN, C. GUESTRIN, 2016)

NATALLIE BAIKEVICH

HARDWARE ACCELERATION FOR DATA PROCESSING SEMINAR

ETH ZÜRICH

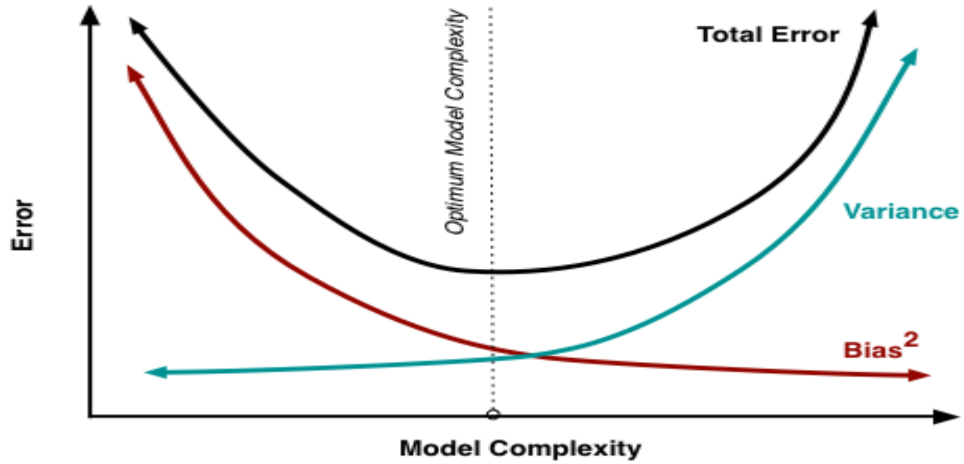
MOTIVATION

- ✓ **Effective** statistical models
- ✓ **Scalable** system
- ✓ **Successful** real-world applications

XGBoost

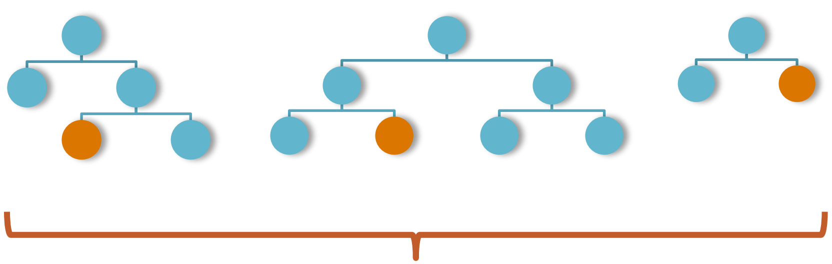
eXtreme Gradient Boosting

BIAS-VARIANCE TRADEOFF



Random Forest

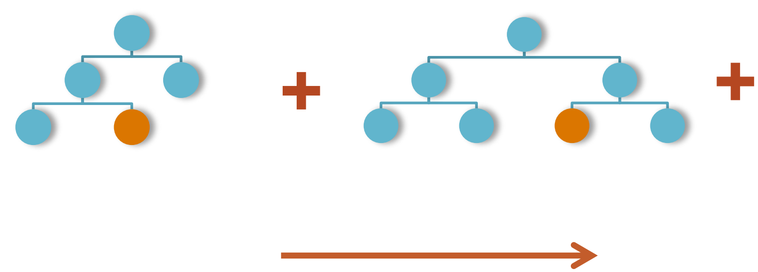
Variance ↓



Voting

Boosting

Bias ↓



A BIT OF HISTORY

AdaBoost, 1996

Random Forests, 1999

Gradient Boosting Machine, 2001

A BIT OF HISTORY

AdaBoost, 1996

Random Forests, 1999

Gradient Boosting Machine, 2001

Various improvements in tree boosting

XGBoost package

A BIT OF HISTORY

AdaBoost, 1996

Random Forests, 1999

Gradient Boosting Machine, 2001

Various improvements in tree boosting

XGBoost package

1st Kaggle success: Higgs Boson Challenge

17/29 winning solutions in 2015

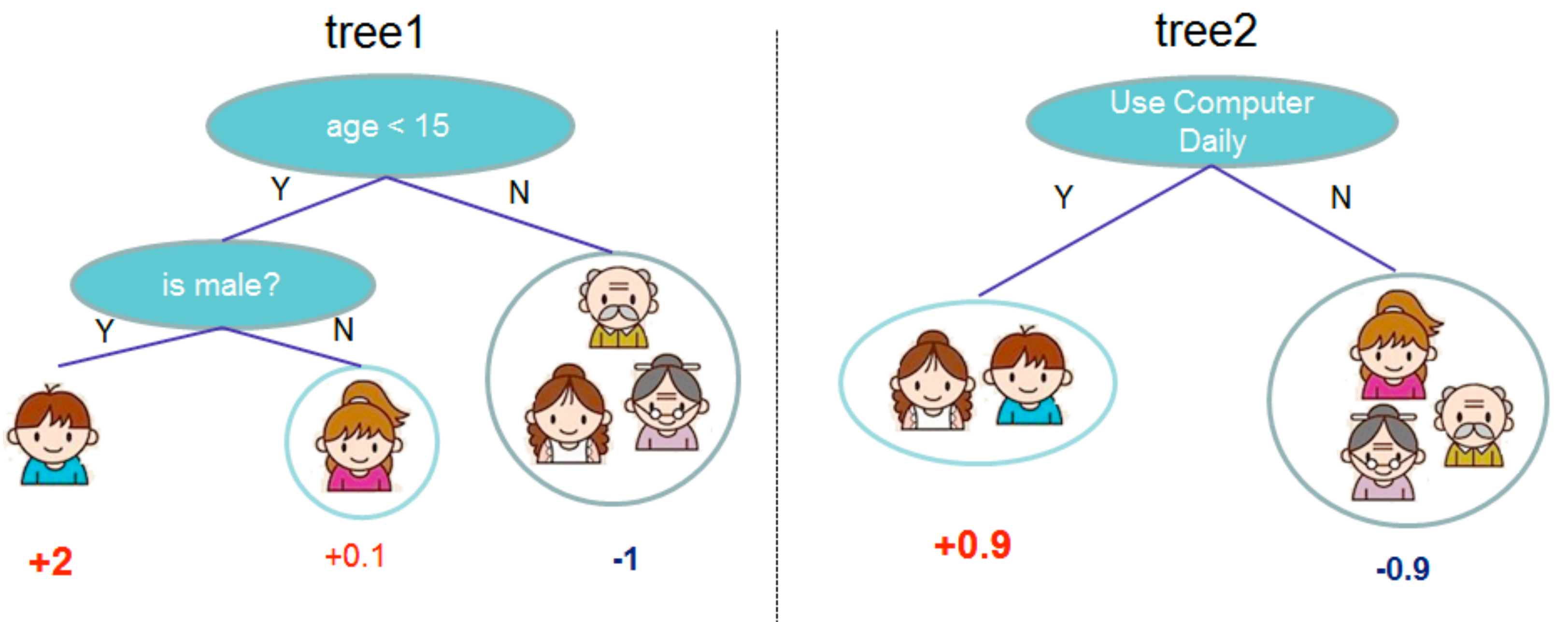
WHY DOES XGBOOST WIN "EVERY" MACHINE LEARNING COMPETITION?

- (MASTER THESIS, D. NIELSEN, 2016)

- Maksims Volkovs, Guangwei Yu and Tomi Poutanen, 1st place of the 2017 ACM RecSys challenge. Link to paper.
- Vlad Sandulescu, Mihai Chiru, 1st place of the KDD Cup 2016 competition. Link to the arxiv paper.
- Marios Michailidis, Mathias Müller and HJ van Veen, 1st place of the Dato Truely Native? competition. Link to the Kaggle interview.
- Vlad Mironov, Alexander Guschin, 1st place of the CERN LHCb experiment Flavour of Physics competition. Link to the Kaggle interview.
- Josef Slavicek, 3rd place of the CERN LHCb experiment Flavour of Physics competition. Link to the Kaggle interview.
- Mario Filho, Josef Feigl, Lucas, Gilberto, 1st place of the Caterpillar Tube Pricing competition. Link to the Kaggle interview.
- Qingchen Wang, 1st place of the Liberty Mutual Property Inspection. Link to the Kaggle interview.
- Chenglong Chen, 1st place of the Crowdflower Search Results Relevance. Link to the winning solution.
- Alexandre Barachant ("Cat") and Rafał Cycoń ("Dog"), 1st place of the Grasp-and-Lift EEG Detection. Link to the Kaggle interview.
- Halla Yang, 2nd place of the Recruit Coupon Purchase Prediction Challenge. Link to the Kaggle interview.
- Owen Zhang, 1st place of the Avito Context Ad Clicks competition. Link to the Kaggle interview.
- Keiichi Kuroyanagi, 2nd place of the Airbnb New User Bookings. Link to the Kaggle interview.
- Marios Michailidis, Mathias Müller and Ning Situ, 1st place Homesite Quote Conversion. Link to the Kaggle interview.

Source: https://github.com/dmlc/xgboost/tree/master/demo#machine-learning-challenge-winning-solutions

TREE ENSEMBLE



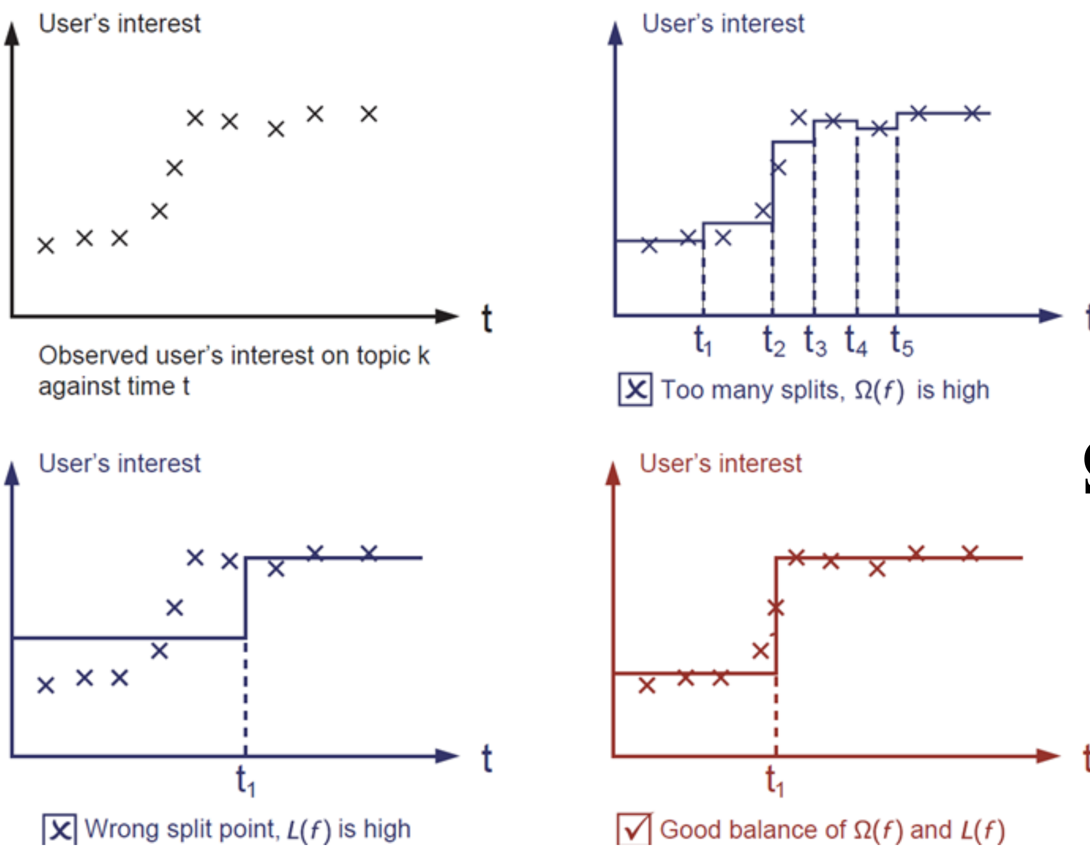
f() = 2 + 0.9= 2.9   f()= -1 - 0.9 = -1.9

REGULARIZED LEARNING OBJECTIVE

loss

$$L = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k)$$

regularization



$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i)$$

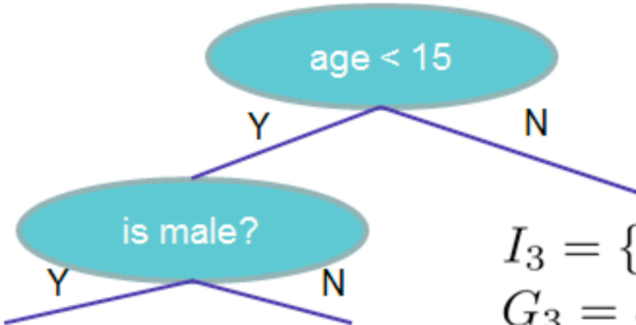
$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \|w\|^2$$

# of leaves

Source: http://xgboost.readthedocs.io/en/latest/model.html

SCORE CALCULATION

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



$I_1 = \{1\}$
$G_1 = g_1$
$H_1 = h_1$

$I_2 = \{4\}$
$G_2 = g_4$
$H_4 = h_4$

$I_3 = \{2, 3, 5\}$
$G_3 = g_2 + g_3 + g_5$
$H_3 = h_2 + h_3 + h_5$

Statistics for each leaf

Score

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

The smaller the score is, the better the structure is

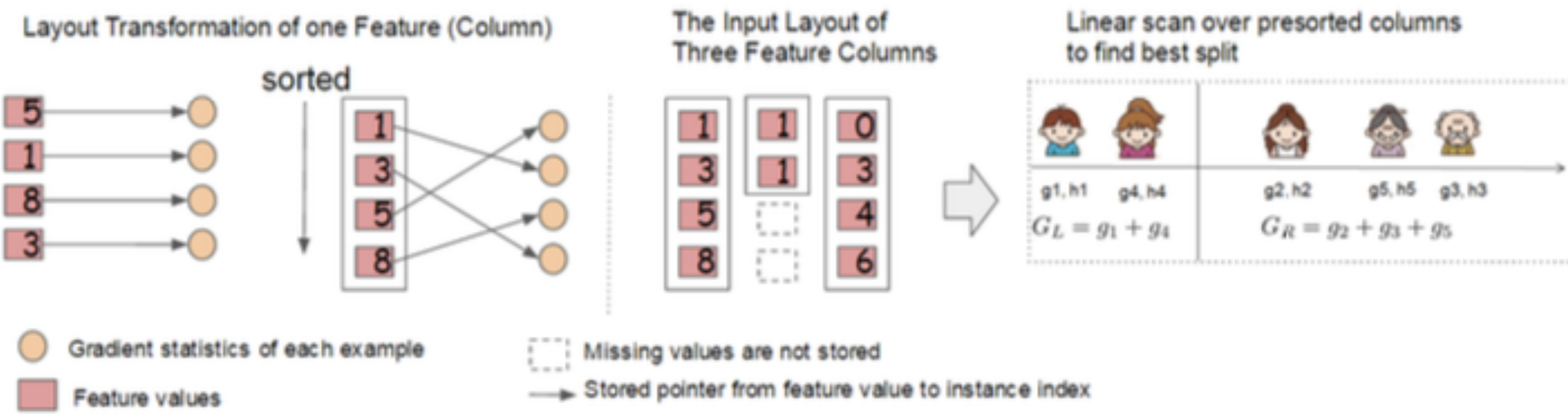
1st order gradient

2nd order gradient

ALGORITHM FEATURES

- ✓ Regularized objective
- ✓ Shrinkage and column subsampling
- ✓ Split finding: exact & approximate, global & local
- ✓ Weighted quantile sketch
- ✓ Sparsity-awareness

SYSTEM DESIGN: BLOCK STRUCTURE



Max depth **Sorted structure –> linear scan**

$$O(Kd\|x\|_0 \log n) \rightarrow O(Kd\|x\|_0 + \|x\|_0 \log B)$$

 # trees **# non-missing entries**

Blocks can be

- ✓ **Distributed across machines**
- ✓ **Stored on disk in out-of-core setting**

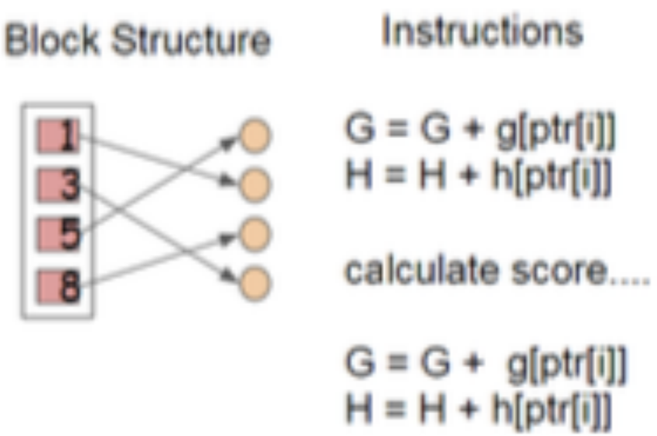
SYSTEM DESIGN: CACHE-AWARE ACCESS

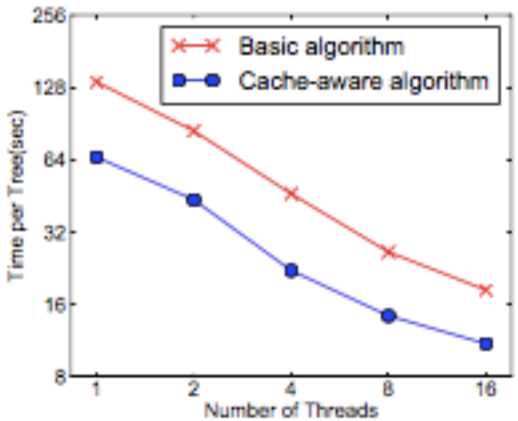
Improved split finding

Non-continuous memory access

✓ Allocate internal buffer

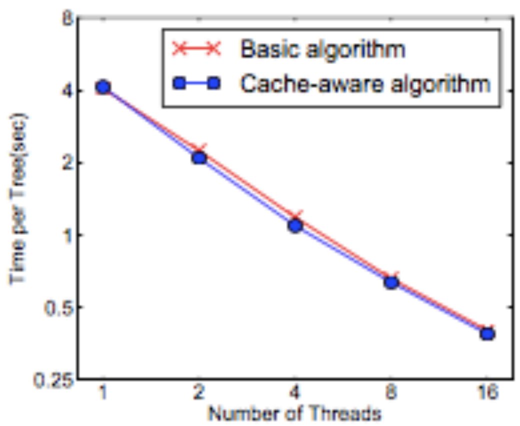
✓ Prefetch gradient statistics





(b) Higgs 10M

Datasets:
Larger vs Smaller



(d) Higgs 1M

SYSTEM DESIGN: BLOCK STRUCTURE

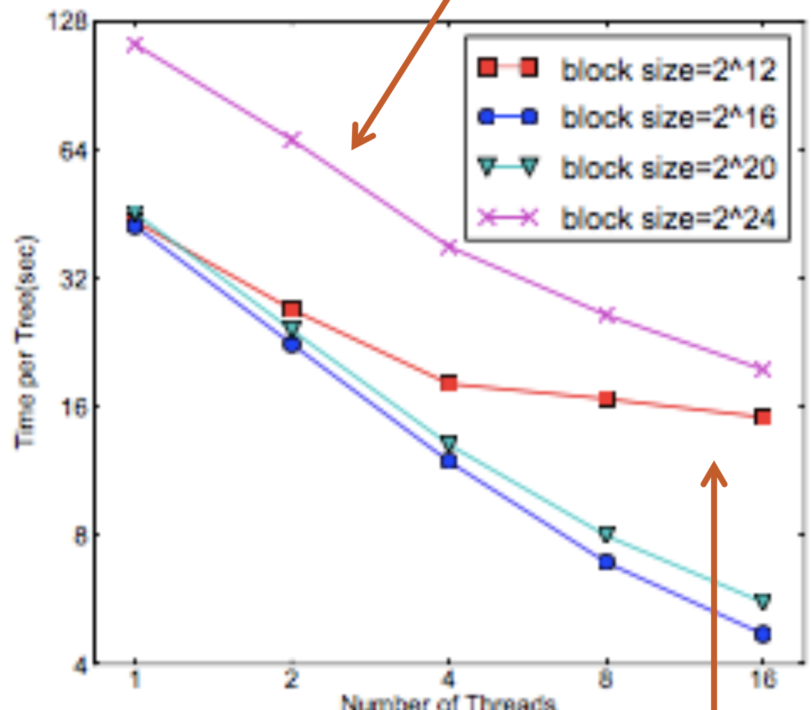
Prefetch
in independent thread

Compression by columns (**CSC**):

Decompression
vs
Disk Reading

Block **sharding**:
Use multiple disks

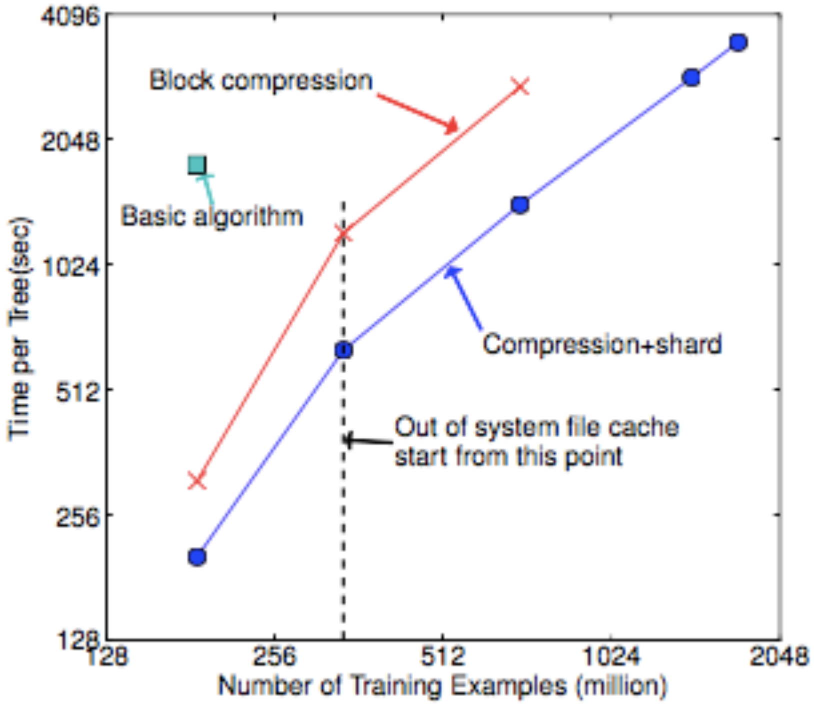
Too large blocks, cache misses

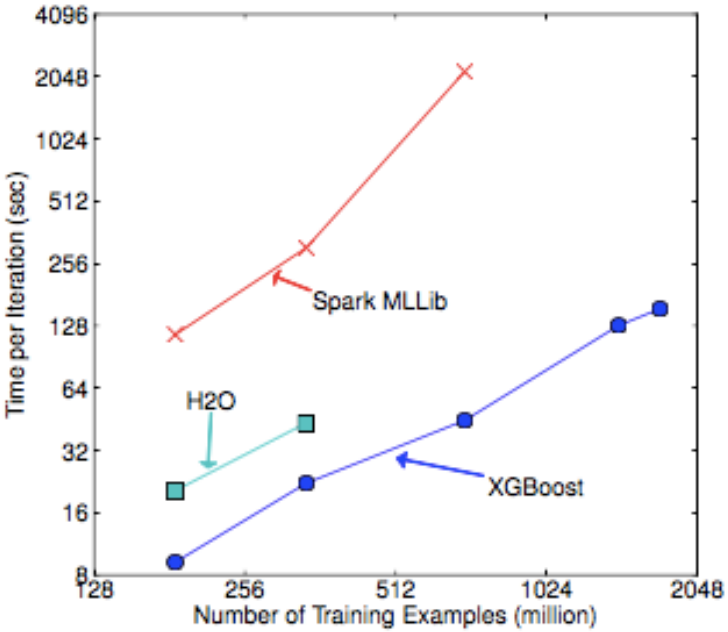


(a) Allstate 10M

Too small, inefficient parallelization

EVALUATION





(b) Per iteration cost exclude data loading

AWS c3.8xlarge machine:
32 virtual cores, 2x320GB SSD,
60 GB RAM

32 m3.2xlarge machines, each:
8 virtual cores, 2x80GB SSD,
30GB RAM

DATASETS

Dataset	n	m	Task
Allstate	10M	4227	Insurance claim classification
Higgs Boson	10M	28	Event classification
Yahoo LTRC	473K	700	Learning to rank
Criteo	1.7B	67	Click through rate prediction

WHAT'S NEXT?

XGBoost

Scalability

Weighted quantiles

Sparsity-awareness

Cache-awarereness

Data compression

→

Tuning

Hyperparameter optimization

Parallel Processing

GPU

FPGA

Model Extensions

DART (+ Dropouts)

LinXGBoost

More Applications

QUICK OVERVIEW

+ Nicely structured paper, easily comprehensible
+ Real framework, widely used for many ML problems
+ Combination of improvements both on model and implementation sides to achieve scalability
+ Reference point for further research in tree boosting,

- The concepts are not that novel themselves
- Does not explain why some of the models are not compared in all experiments
- Is the compression efficient for dense datasets?
- What if there's a lot of columns rather than rows (e.g. medical data)?

THANK YOU!