Stochastic Gradient Boosted Distributed Decision Trees

A study headed by Yahoo! labs
Presented by Shiran Dudy
18/04/14

Outline

- GBDT
- Distributing the GBDT Algorithm
- MPI and MapReduce implementation
- Experiments
- Results
- Discussion

Gradient Boosted Distributed Tree(GBDT)

what is it?

Boosting

ensemble technique in which learners are learned sequentially with early learners fitting a simple model to the data and analyzing the data for errors - and later models focus on these errors trying to get them right. In the end all learners are given weights and combined to create an overall predictor.

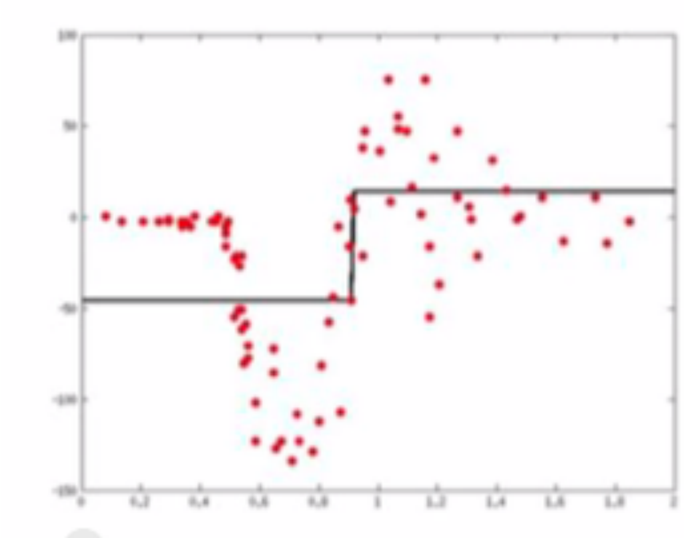
https://www.youtube.com/watch?v=sRktKszFmSk, Ensembles (3): Gradient Boosting, Alexander Ihler

Gradient Boosted Distributed Tree(GBDT)

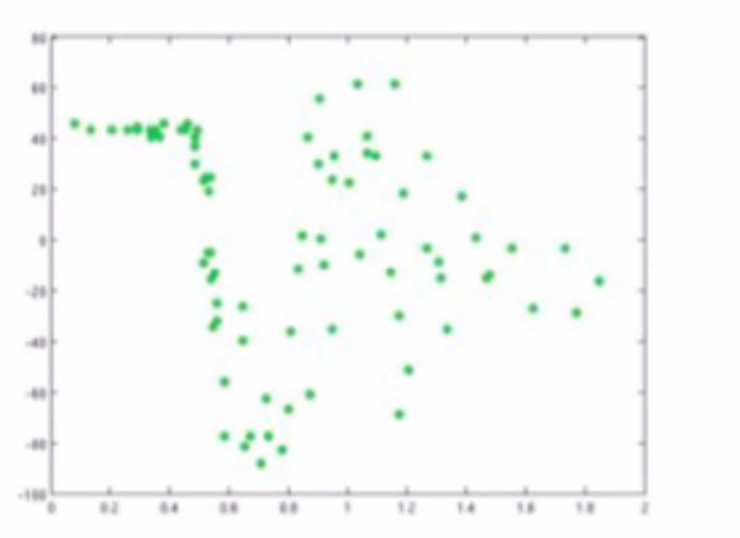
what is it?

- learn a regression predictor
- compute the error residual
- learn to predict the residual

learn a simple predictor



Then try to correct its errors



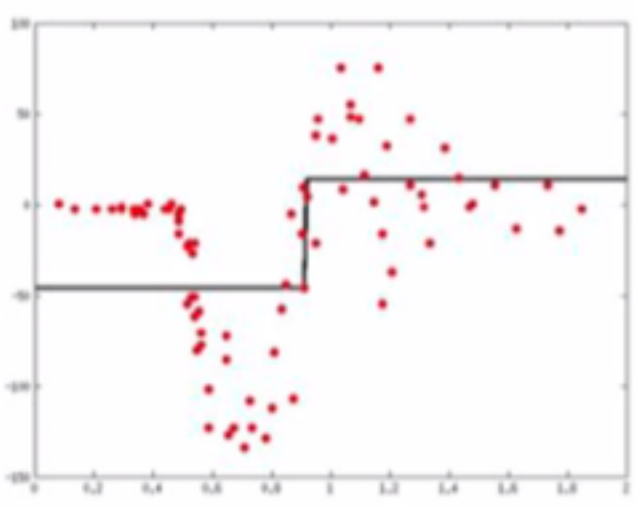
https://www.youtube.com/watch?v=sRktKszFmSk, Ensembles (3): Gradient Boosting, Alexander Ihler

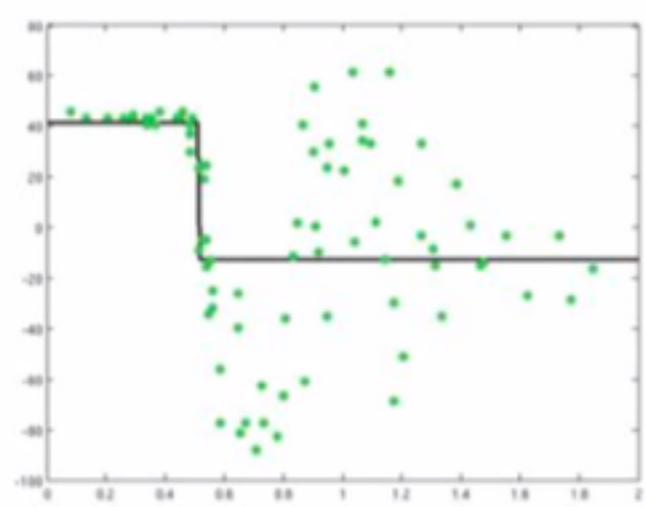
Gradient Boosted Distributed Tree(GBDT)

what is it?

- learn a regression predictor
- compute the error residual
- learn to predict the residual

learn a simple predictor   Then try to correct its errors





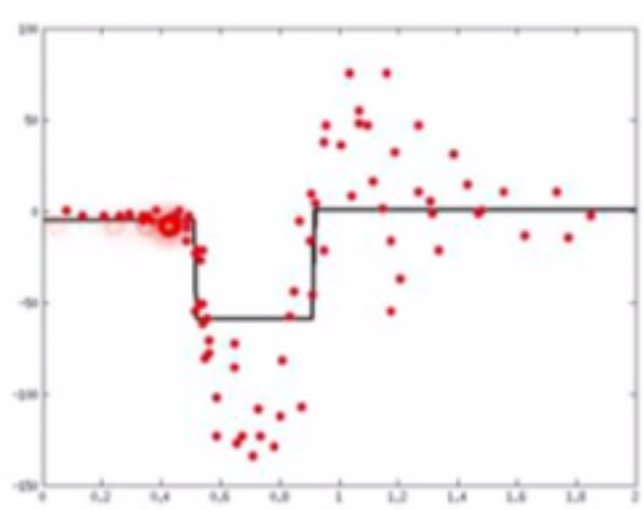
https://www.youtube.com/watch?v=sRktKszFmSk, Ensembles (3): Gradient Boosting, Alexander Ihler

Gradient Boosted Distributed Tree(GBDT)

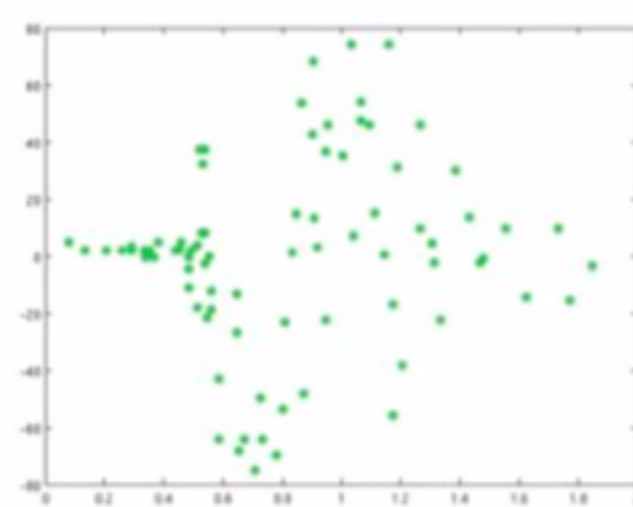
what is it?

- learn a regression predictor
- compute the error residual
- learn to predict the residual

combining gives a better predictor



can try to correct its errors also and repeat



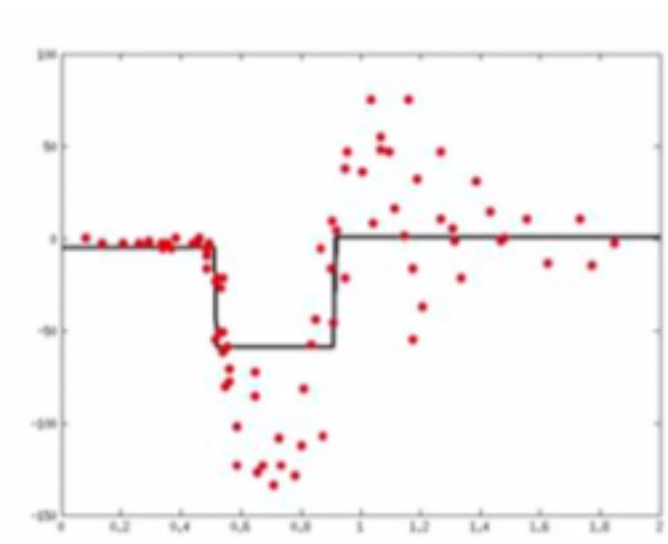
https://www.youtube.com/watch?v=sRktKszFmSk, Ensembles (3): Gradient Boosting, Alexander Ihler

Gradient Boosted Distributed Tree(GBDT)

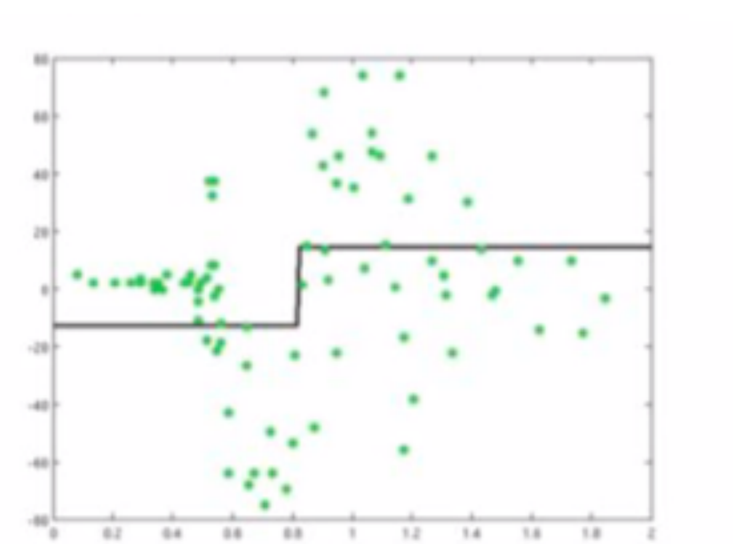
what is it?

- learn a regression predictor
- compute the error residual
- learn to predict the residual

combining gives a better predictor



can try to correct its errors also and repeat

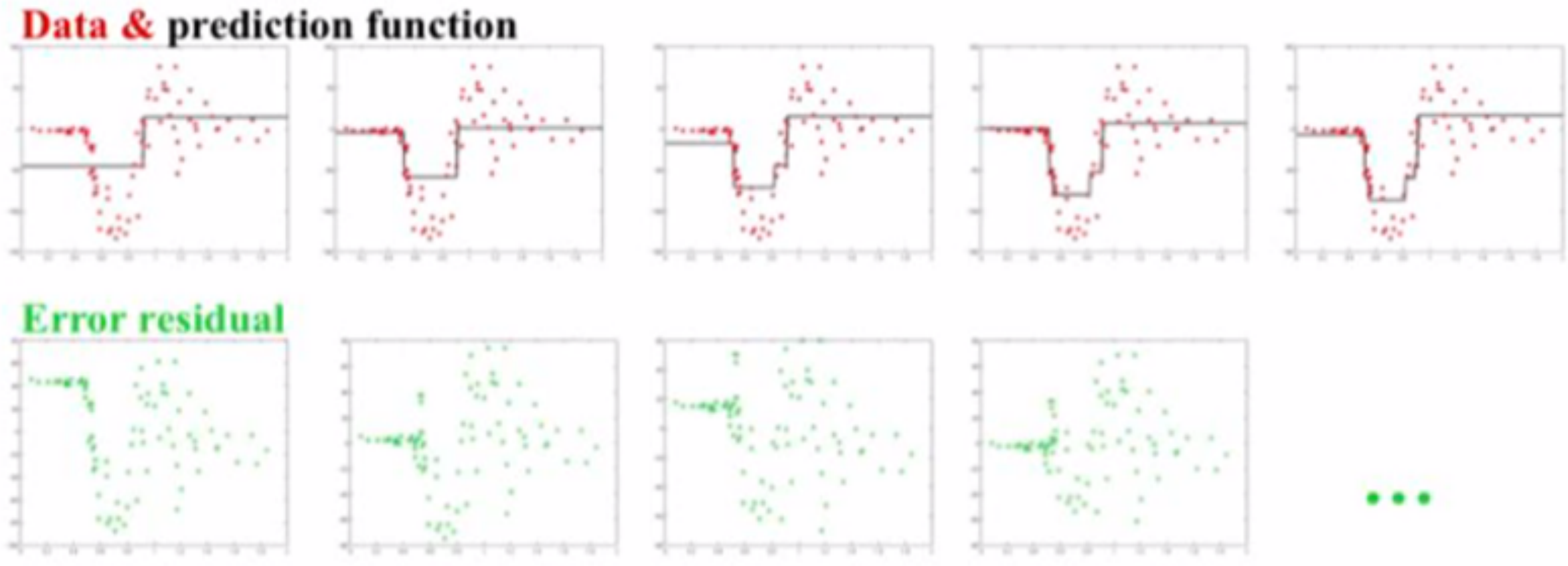


https://www.youtube.com/watch?v=sRktKszFmSk, Ensembles (3): Gradient Boosting, Alexander Ihler

Gradient Boosted Distributed Tree(GBDT)

what is it?

- learn a regression predictor
- compute the error residual
- learn to predict the residual



https://www.youtube.com/watch?v=sRktKszFmSk, Ensembles (3): Gradient Boosting, Alexander Ihler

The Goal

Distributing the GBDT Algorithm

The Goal

Distributing the GBDT Algorithm

The Goal

But WHY?

There's a need to incorporate increasing numbers of features and instances in training data and because existing methods require all training data to be in physical memory

Distributing the GBDT Algorithm

The Goal

HOW?

By improving the training time of individual trees and not on parallelizing the actual boosting phase

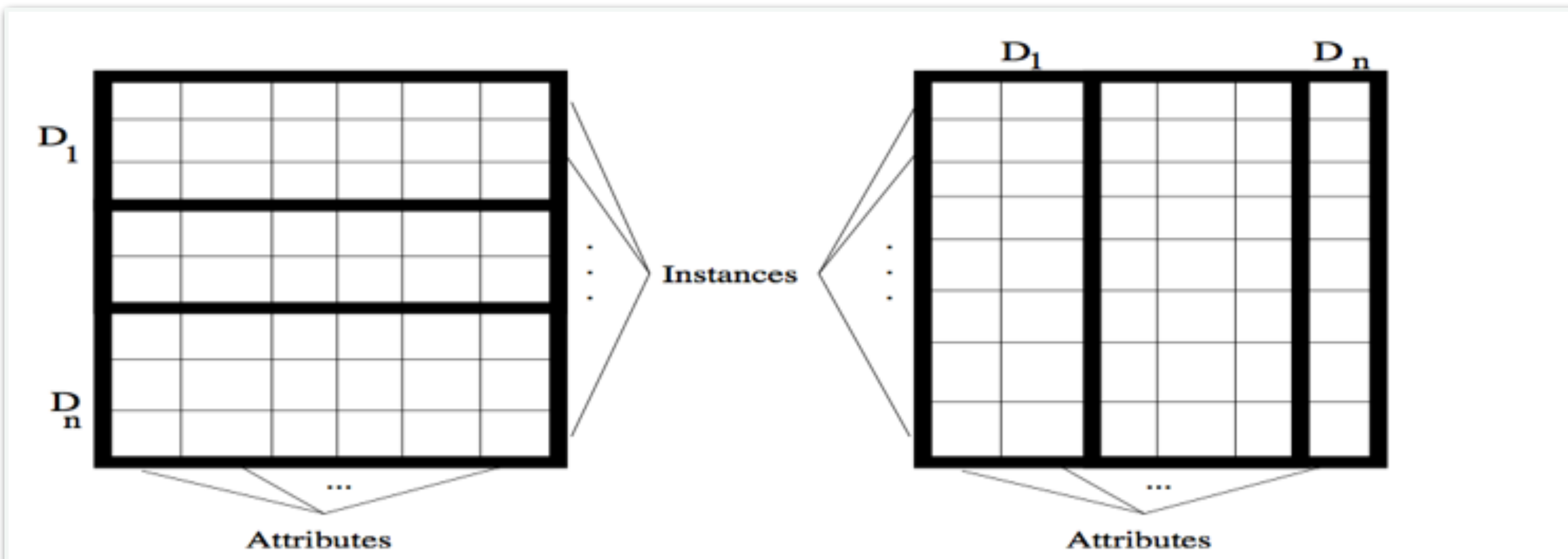
MapReduce

MPI

Distributing the GBDT Algorithm

The Goal

HOW to Partition the training data?



CARAGEA,D.@ I A framework for learning from distributed data using sufficient statistics and its application to learning decision trees,2004

Distributing the GBDT Algorithm

The Goal

HOW to Partition the training data?

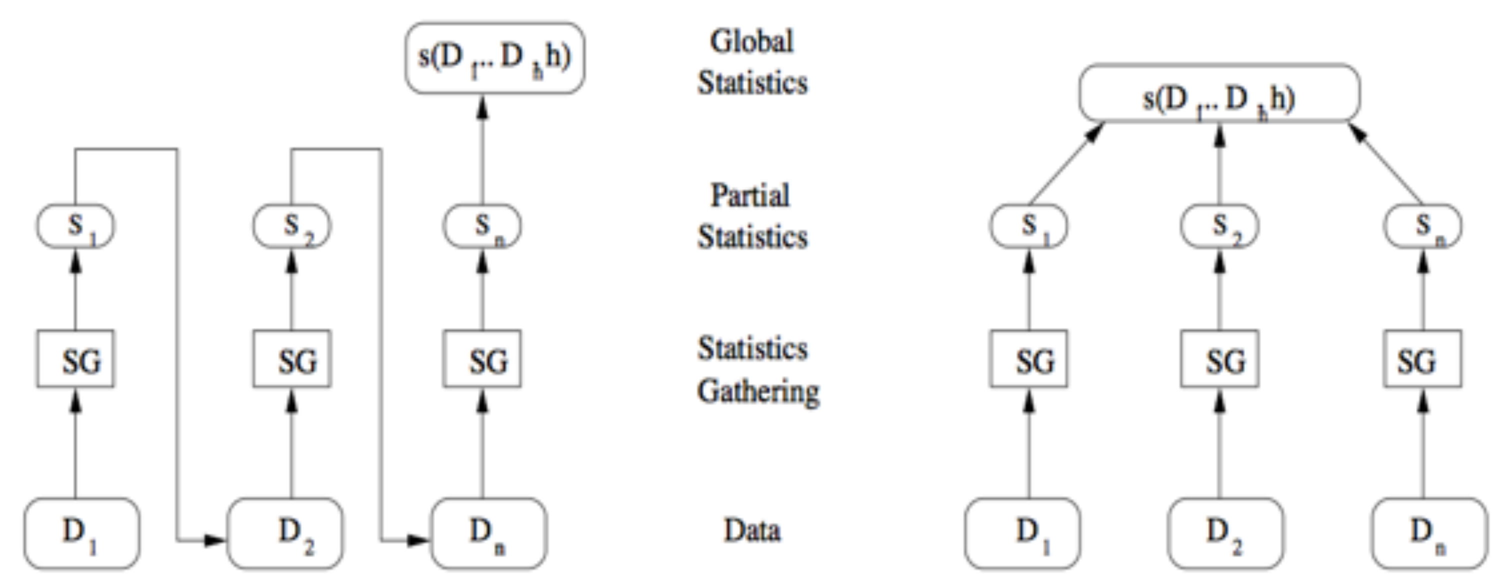


Figure 4. Distributed Statistics Gathering: (Left) Serial. (Right) Parallel.

CARAGEA,D.@ I A framework for learning from distributed data using sufficient statistics and its application to learning decision trees,2004

Distributing the GBDT Algorithm

MapReduce

Algorithm 1 Aggregating candidate splits

```
map(key, value):
F ⇐ set of features
sample ⇐ split(value,delim)
for f in F do
    key = (f, sample[f])
    value = (sample[residual], sample[weight])
    emit(key, value)
end for

reduce(key, values):
residual_sum ⇐ 0
weight_sum ⇐ 0
for v in values do
    residual_sum ⇐ residual_sum + v.residual
    weight_sum ⇐ weight_sum + v.weight
end for
emit(key, (residual_sum,weight_sum))
```

Stochastic Gradient Boosted Distributed Decision Trees, Yahoo! labs, 2009

Distributing the GBDT Algorithm

MapReduce

Algorithm 2 Partitioning a Node n

```
map(key, value):
sample ⇐ split(value,delim)
if sample[n.feature] < n.splitpoint then
    residual = sample[residual]+ n.left_response
else
    residual = sample[residual]+ n.right_response
end if
emit(key, value)
```

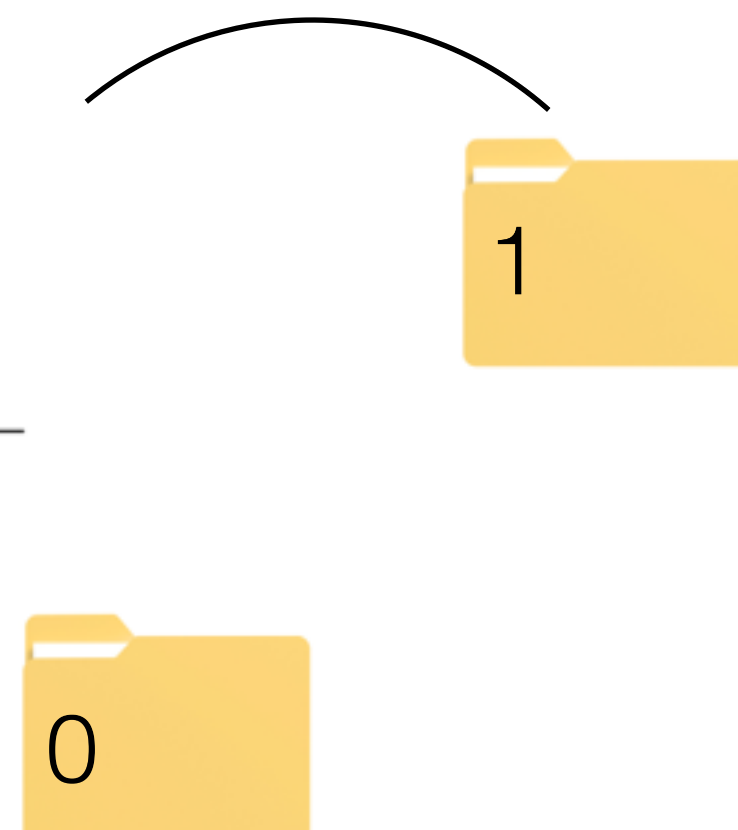
Stochastic Gradient Boosted Distributed Decision Trees, Yahoo! labs, 2009

Distributing the GBDT Algorithm

MapReduce

Algorithm 2 Partitioning a Node n

```
map(key, value):
sample ⇐ split(value,delim)
if sample[n.feature] < n.splitpoint then
    residual = sample[residual]+ n.left_response
else
    residual = sample[residual]+ n.right_response
end if
emit(key, value)
```



additional communication cost caused by writing out multiple files when splitting a node—-> high system overhead

Stochastic Gradient Boosted Distributed Decision Trees, Yahoo! labs, 2009

Message Passing Interface (MPI)

What is it?

a parallel MPI program is launched as sperate processes (tasks), each with their own address space -> it requires partitioning data across tasks

a task accesses the data of another task through a transaction called “message passing” in which a copy of the data (message) is transferred (passed) from one task to another

Stochastic Gradient Boosted Distributed Decision Trees, Yahoo! labs, 2009

Message Passing Interface (MPI)

Process

$$S^{'}_{i,j} = \textbf{argmax}_{i,j}\{gain(c_{i,j})\}$$

Stochastic Gradient Boosted Distributed Decision Trees, Yahoo! labs, 2009

Message Passing Interface (MPI)

Process

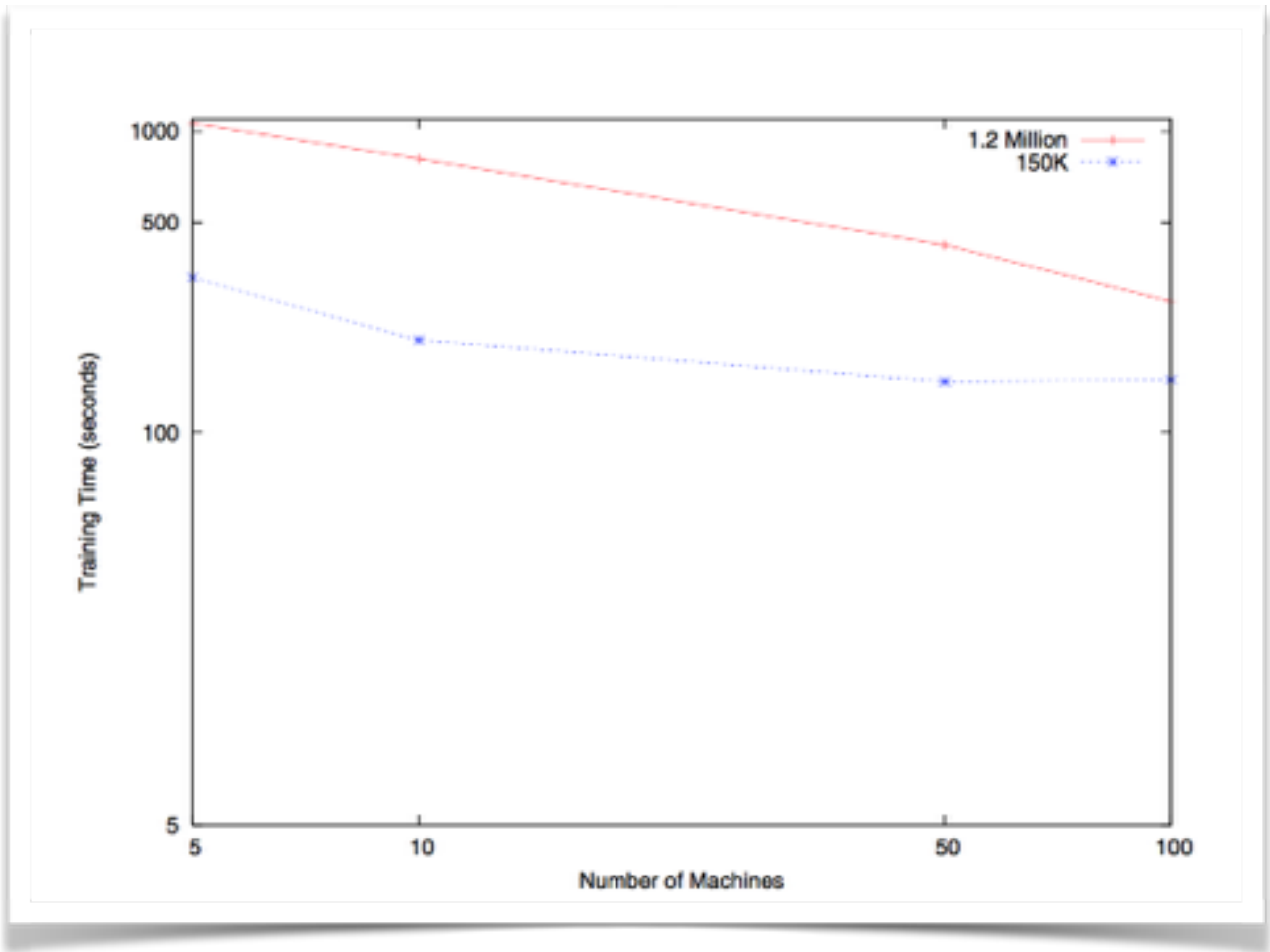
$$S'_{i,j} = \mathbf{argmax}_{i,j}\{gain(c_{i,j})\}$$

each machine is given a subset of the feature space and can compute the best local split for its j's and i's and sends her result to her friends when everybody knows the best cut

Stochastic Gradient Boosted Distributed Decision Trees, Yahoo! labs, 2009

Experiment Results

MapReduce

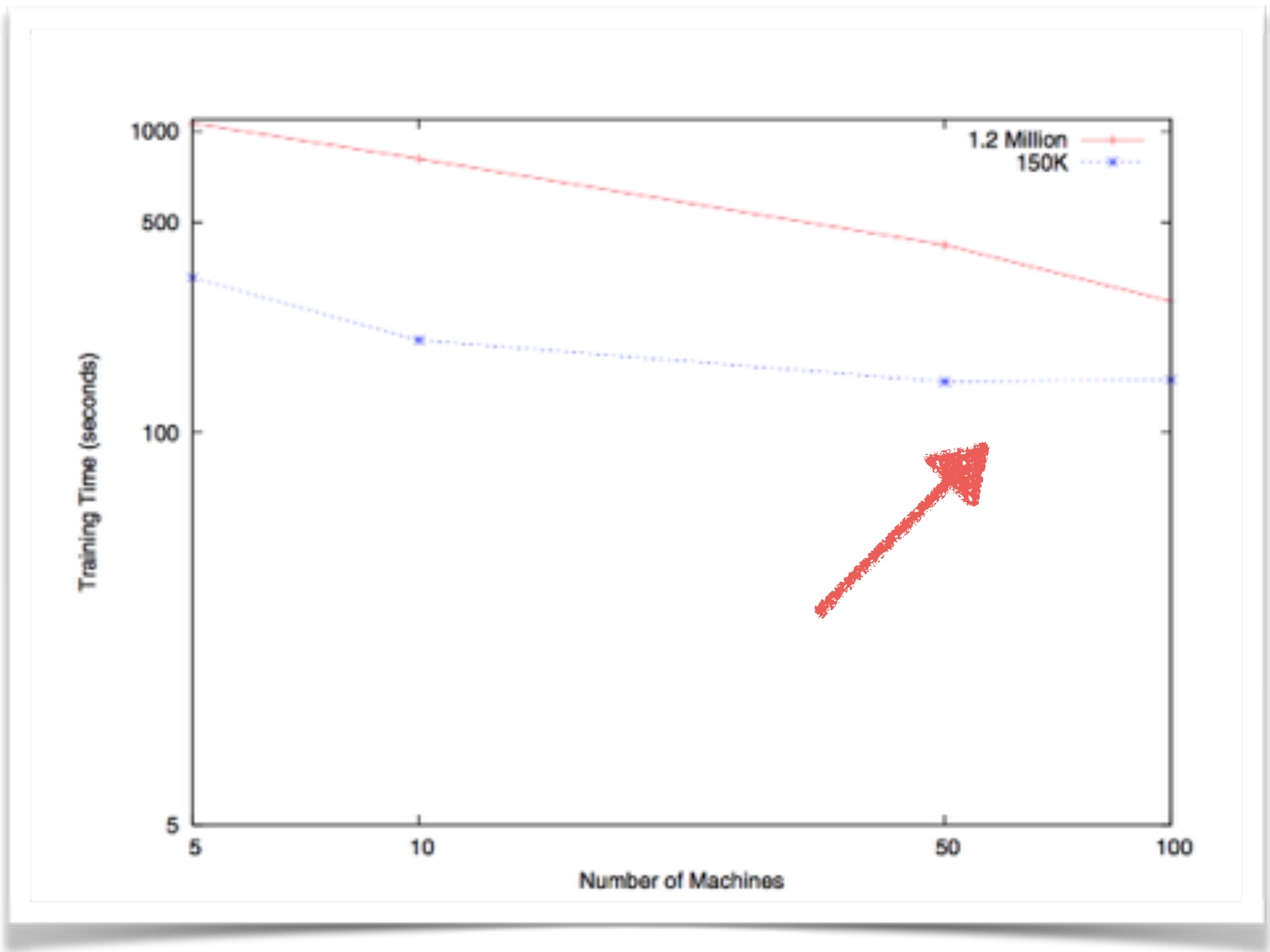


Stochastic Gradient Boosted Distributed Decision Trees, Yahoo! labs, 2009

Experiment Results

MapReduce

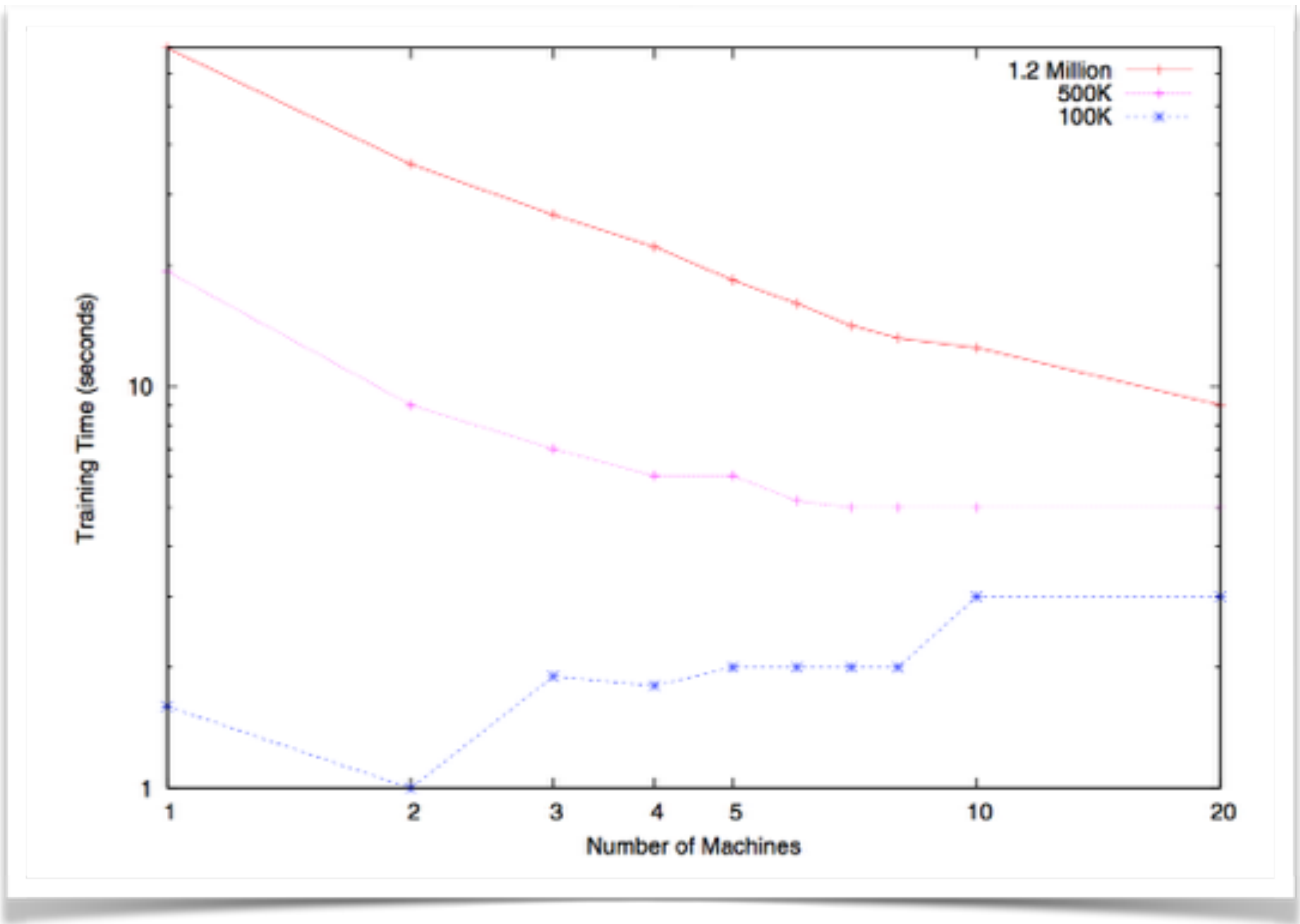
communication overhead. not as good even in comparison to non parallel implementation



Stochastic Gradient Boosted Distributed Decision Trees, Yahoo! labs, 2009

Experiment Results

MPI

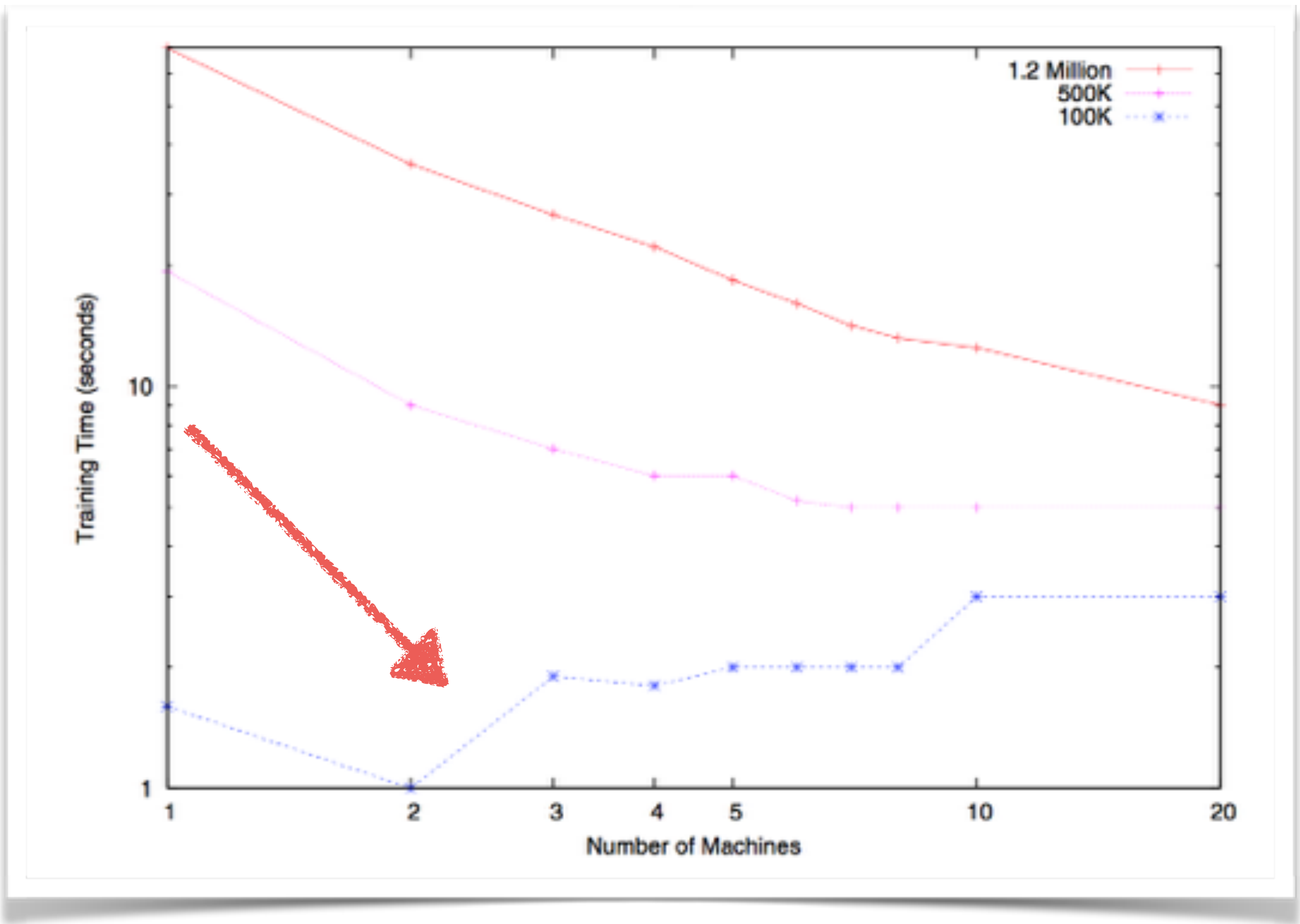


Stochastic Gradient Boosted Distributed Decision Trees, Yahoo! labs, 2009

Experiment Results

MPI

for 100k the overhead was too high to be useful

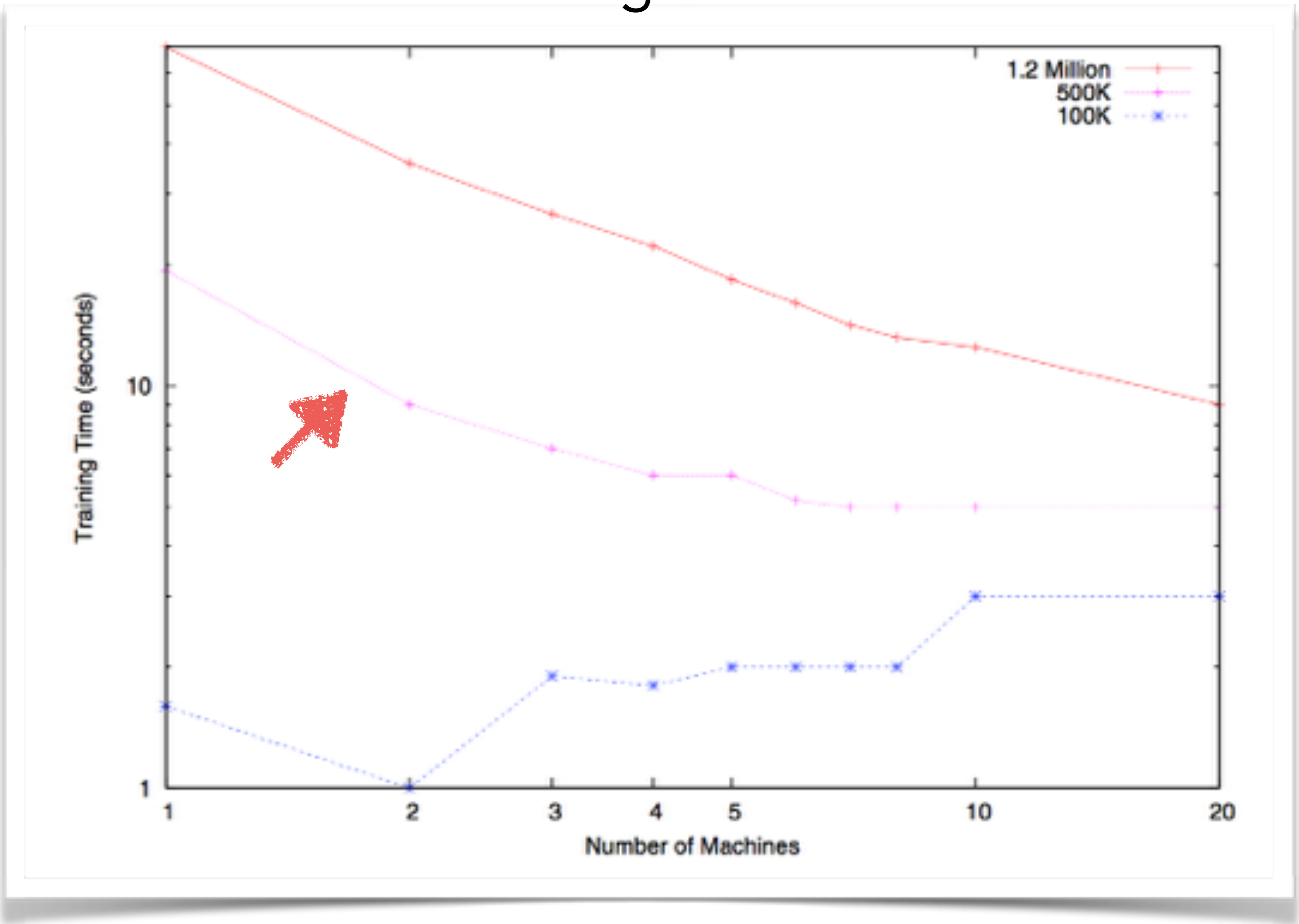


Stochastic Gradient Boosted Distributed Decision Trees, Yahoo! labs, 2009

Experiment Results

MPI

training time was reduced in 0.5 after using 2 machines and continue to improve until 5

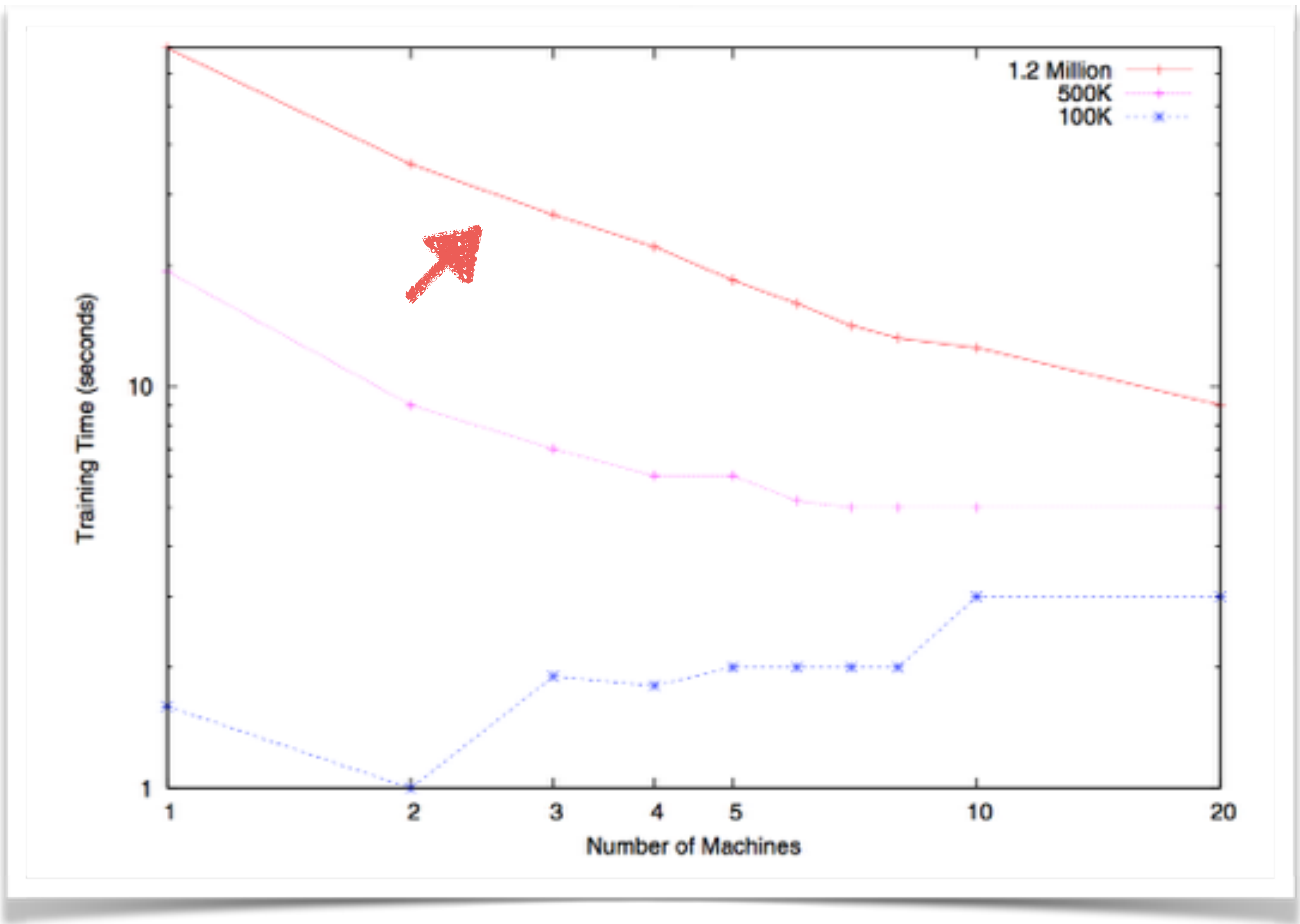


Stochastic Gradient Boosted Distributed Decision Trees, Yahoo! labs, 2009

Experiment Results

MPI

an improvement from 70 sec to 9 sec per tree



Stochastic Gradient Boosted Distributed Decision Trees, Yahoo! labs, 2009

Intake

MapReduce

limited amount of code :)

high scalability :)

communication cost :(

MPI

high scalability :)

communication cost :)

overall -good :)

Matlab for GBDT

code

```
% Data set  X, Y
mu = mean(Y);     % Often start with constant "mean" predictor
dY = Y - mu;      %   subtract this prediction away
For k=1:Nboost,
  Learner{k} = Train_Regressor(X,dY);
  alpha(k) = 1;  % alpha: a "learning rate" or "step size"
  % smaller alphas need to use more classifiers, but tend to
  %   predict better given enough of them

  % compute the residual given our new prediction
  dY = dY - alpha(k) * predict(Learner{k}, X)
end;


% Test data Xtest
[Ntest,D] = size(Xtest);
predict = zeros(Ntest,1);          % Allocate space
For k=1:Nboost,                    % Predict with each learner
  predict = predict + alpha(k)*predict(Learner{k}, Xtest);
end;
```

https://www.youtube.com/watch?v=sRktKszFmSk, Ensembles (3): Gradient Boosting, Alexander Ihler