

# PLANET: Massively Parallel Learning of Tree Ensembles with MapReduce

Luyu Wang  
SciCom Group, UW

UNIVERSITY OF  
**WATERLOO**



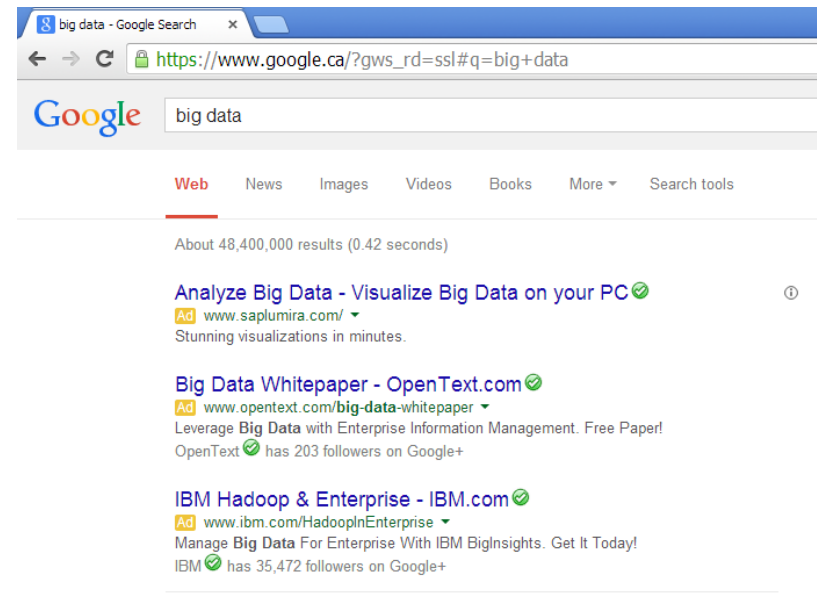
# Content

- Background
- Methodology
- Results
- Conclusion



# Google's Bounce Rate Prediction Problem

- 'Bounce'

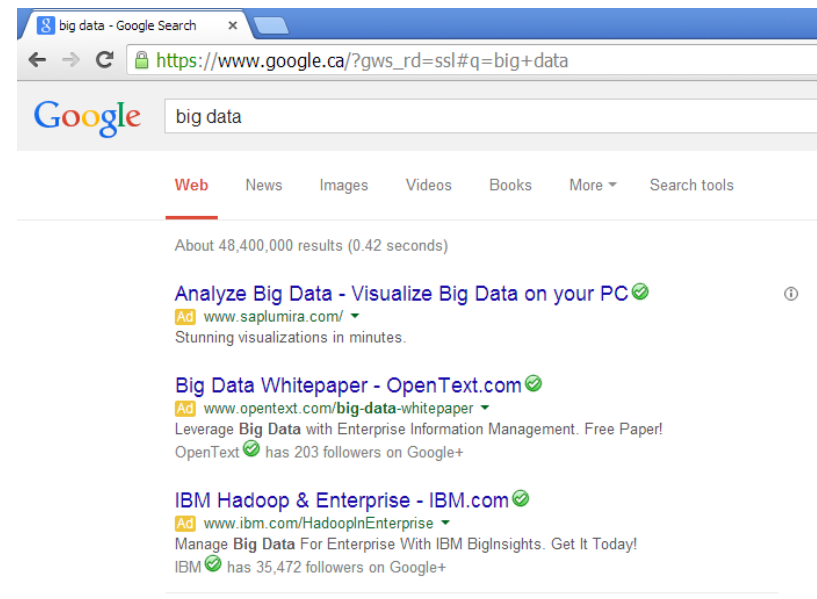


The screenshot shows a Google search results page for the query "big data". The browser's address bar displays "https://www.google.ca/?gws\_rd=ssl#q=big+data". The search bar contains the text "big data". Below the search bar, there are navigation tabs for "Web", "News", "Images", "Videos", "Books", "More", and "Search tools". The search results indicate "About 48,400,000 results (0.42 seconds)". The first three results are advertisements:

- Analyze Big Data - Visualize Big Data on your PC** (Ad) from [www.saplumira.com/](http://www.saplumira.com/). Description: "Stunning visualizations in minutes."
- Big Data Whitepaper - OpenText.com** (Ad) from [www.opentext.com/big-data-whitepaper](http://www.opentext.com/big-data-whitepaper). Description: "Leverage Big Data with Enterprise Information Management. Free Paper! OpenText has 203 followers on Google+."
- IBM Hadoop & Enterprise - IBM.com** (Ad) from [www.ibm.com/HadoopnEnterprise](http://www.ibm.com/HadoopnEnterprise). Description: "Manage Big Data For Enterprise With IBM BigInsights. Get It Today! IBM has 35,472 followers on Google+."

# Google's Bounce Rate Prediction Problem

- 'Bounce'
- High bounce rate = poor user experience

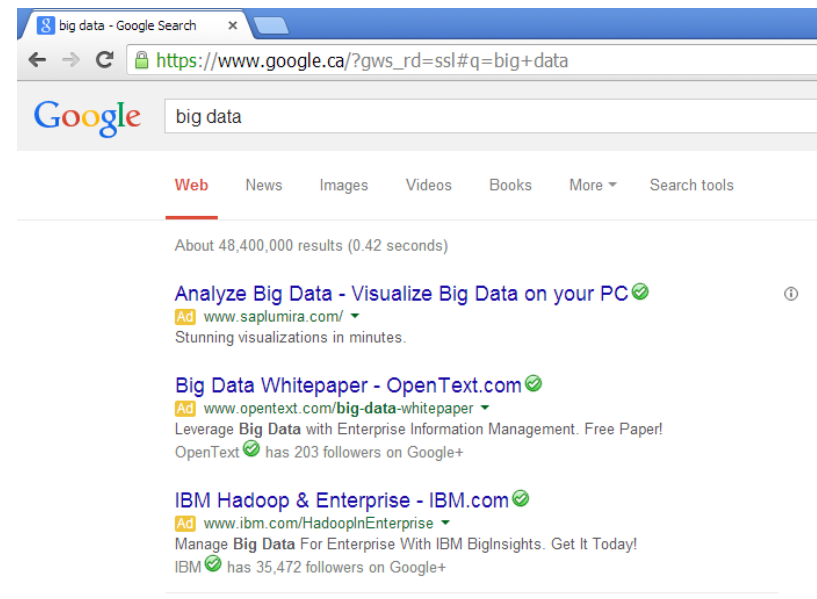


The screenshot shows a Google search results page for the query "big data". The browser's address bar displays the URL "https://www.google.ca/?gws\_rd=ssl#q=big+data". The search bar contains the text "big data". Below the search bar, there are navigation tabs for "Web", "News", "Images", "Videos", "Books", "More", and "Search tools". The "Web" tab is selected. The search results indicate "About 48,400,000 results (0.42 seconds)". The first three results are advertisements:

- Analyze Big Data - Visualize Big Data on your PC** (Ad) from [www.saplumira.com/](http://www.saplumira.com/). Description: "Stunning visualizations in minutes."
- Big Data Whitepaper - OpenText.com** (Ad) from [www.opentext.com/big-data-whitepaper](http://www.opentext.com/big-data-whitepaper). Description: "Leverage Big Data with Enterprise Information Management. Free Paper! OpenText has 203 followers on Google+."
- IBM Hadoop & Enterprise - IBM.com** (Ad) from [www.ibm.com/HadoopnEnterprise](http://www.ibm.com/HadoopnEnterprise). Description: "Manage Big Data For Enterprise With IBM BigInsights. Get It Today! IBM has 35,472 followers on Google+."

# Google's Bounce Rate Prediction Problem

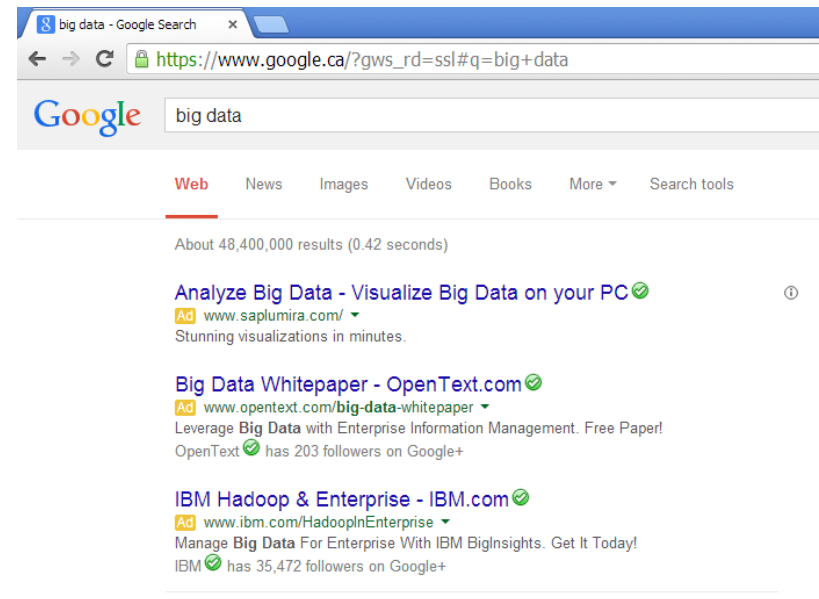
- 'Bounce'
- High bounce rate = poor user experience
- **Task**: to predict bounce rate with data on hand



# Google's Bounce Rate Prediction Problem

- 'Bounce'
- High bounce rate = poor user experience
- **Task**: to predict bounce rate with data on hand

**COMPUTATIONAL  
ADVERTISING**



The screenshot shows a Google search results page for the query 'big data'. The browser address bar displays 'https://www.google.ca/?gws\_rd=ssl#q=big+data'. The search results are filtered to the 'Web' tab and show approximately 48,400,000 results in 0.42 seconds. Three sponsored search results are visible:

- Analyze Big Data - Visualize Big Data on your PC** (Ad) from [www.saplumira.com/](http://www.saplumira.com/). Description: 'Stunning visualizations in minutes.'
- Big Data Whitepaper - OpenText.com** (Ad) from [www.opentext.com/big-data-whitepaper](http://www.opentext.com/big-data-whitepaper). Description: 'Leverage Big Data with Enterprise Information Management. Free Paper! OpenText has 203 followers on Google+'.
- IBM Hadoop & Enterprise - IBM.com** (Ad) from [www.ibm.com/HadoopInEnterprise](http://www.ibm.com/HadoopInEnterprise). Description: 'Manage Big Data For Enterprise With IBM BigInsights. Get It Today! IBM has 35,472 followers on Google+'.

# The Data Mining Tasks

- Discovering patterns in large data sets (knowledge)
- Data mining vs machine learning?
  - Lines are blurred

# The Data Mining Tasks

- Supervised learning
    - Classification
    - Regression
  - Unsupervised learning
    - Clustering
    - Compression
    - Outlier detection
  - Reinforce learning
- .....

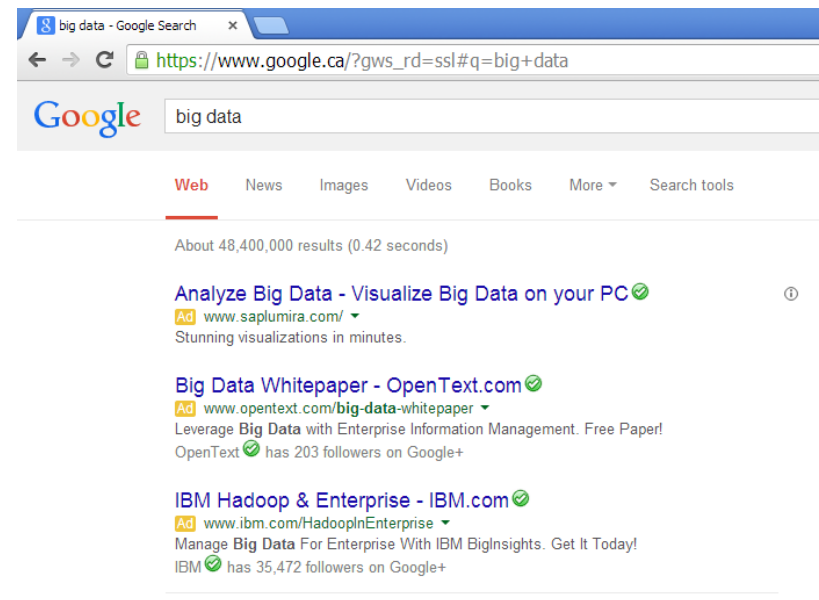


# The Data Mining Tasks

- **Supervised learning**
  - Classification
  - **Regression**
- **Unsupervised learning**
  - Clustering
  - Compression
  - Outlier detection
- **Reinforce learning**
  - .....

# Google's Bounce Rate Prediction Problem

- Go back to 'bounce'



The screenshot shows a Google search results page for the query "big data". The browser's address bar displays the URL "https://www.google.ca/?gws\_rd=ssl#q=big+data". The search results are filtered to the "Web" tab. The top result is an advertisement for "Analyze Big Data - Visualize Big Data on your PC" from www.saplumira.com, with the text "Stunning visualizations in minutes." Below it is another advertisement for "Big Data Whitepaper - OpenText.com" from www.opentext.com, with the text "Leverage Big Data with Enterprise Information Management. Free Paper! OpenText has 203 followers on Google+." The third result is an advertisement for "IBM Hadoop & Enterprise - IBM.com" from www.ibm.com/HadoopInEnterprise, with the text "Manage Big Data For Enterprise With IBM BigInsights. Get It Today! IBM has 35,472 followers on Google+."



big data - Google Search x




← → ↻ [https://www.google.ca/?gws\\_rd=ssl#q=big+data](https://www.google.ca/?gws_rd=ssl#q=big+data)




Google big data

Web News Images Videos Books More Search tools

About 48,400,000 results (0.42 seconds)

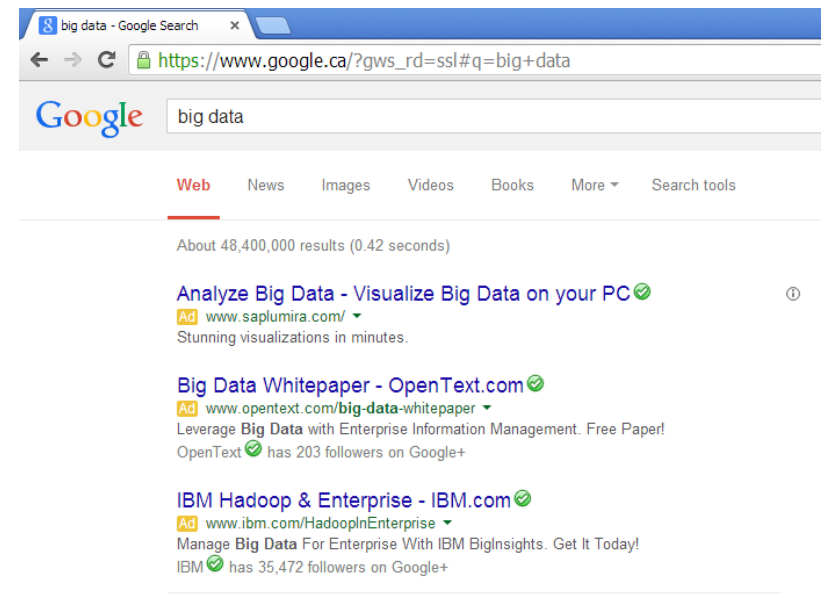
**Analyze Big Data - Visualize Big Data on your PC**  ⓘ  
**Ad** [www.saplumira.com/](http://www.saplumira.com/)   
Stunning visualizations in minutes.

**Big Data Whitepaper - OpenText.com**  ⓘ  
**Ad** [www.opentext.com/big-data-whitepaper](http://www.opentext.com/big-data-whitepaper)   
Leverage **Big Data** with Enterprise Information Management. Free Paper!  
OpenText  has 203 followers on Google+

**IBM Hadoop & Enterprise - IBM.com**  ⓘ  
**Ad** [www.ibm.com/HadoopInEnterprise](http://www.ibm.com/HadoopInEnterprise)   
Manage **Big Data** For Enterprise With IBM BigInsights. Get It Today!  
IBM  has 35,472 followers on Google+

# Google's Bounce Rate Prediction Problem

- Go back to 'bounce'
- One Click:
  - 6 attributes
  - 1 label



# Google's Bounce Rate Prediction Problem

- 6 attributes
  - search query of the click
  - advertiser chosen keyword
  - ad text
  - estimated clickthrough rate of the ad click
  - numeric similarity score
  - whether the ad matches the query
- 1 label
  - bounce or not

# Supervised Learning - Data Model

- Set of attributes

$$\mathcal{X} = \{X_1, X_2, \dots, X_N\}$$

- Output

$$Y$$

- Training data set (the  $i^{\text{th}}$  vector)

$$D^* = \{(\mathbf{x}_i, y_i) \mid \mathbf{x}_i \in D_{x_1} \times D_{x_2} \times \dots \times D_{x_N}\}$$

# Supervised Learning - Task

- Given the training dataset

$$D^* = \{(\mathbf{x}_i, y_i) \mid \mathbf{x}_i \in D_{x_1} \times D_{x_2} \times \dots \times D_{x_N}\}$$

- Goal: to learn a mapping model

$$F : D_{x_1} \times D_{x_2} \times \dots \times D_{x_N} \rightarrow D_Y$$

# Google's Bounce Rate Prediction Problem

- Given 6 attributes
  - search query of the click
  - advertiser chosen keyword
  - ad text
  - estimated clickthrough rate of the ad click
  - numeric similarity score
  - whether the ad matches the query
- Want to know if it is going to be a bounce?

# Supervised Learning - Task

- Given the training dataset

$$D^* = \{(\mathbf{x}_i, y_i) \mid \mathbf{x}_i \in D_{x_1} \times D_{x_2} \times \dots \times D_{x_N}\}$$

- Goal: to learn a mapping model

$$F : D_{x_1} \times D_{x_2} \times \dots \times D_{x_N} \rightarrow D_Y$$

- Tree models
  - Capable of modeling complex tasks

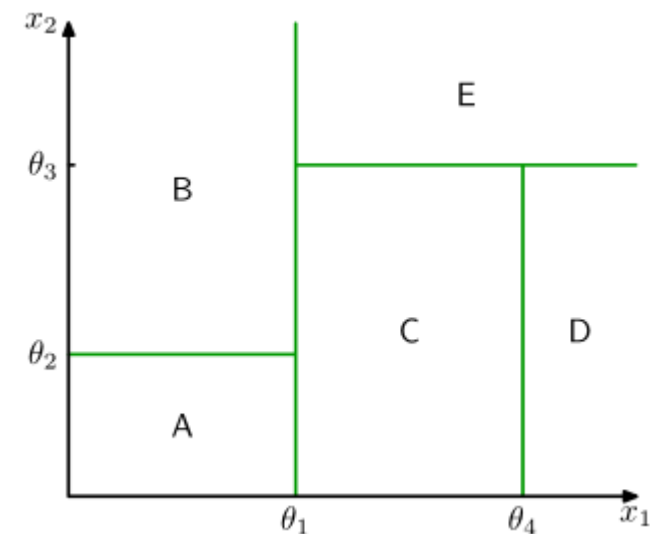


# Tree Model

- Goal: to learn a mapping model

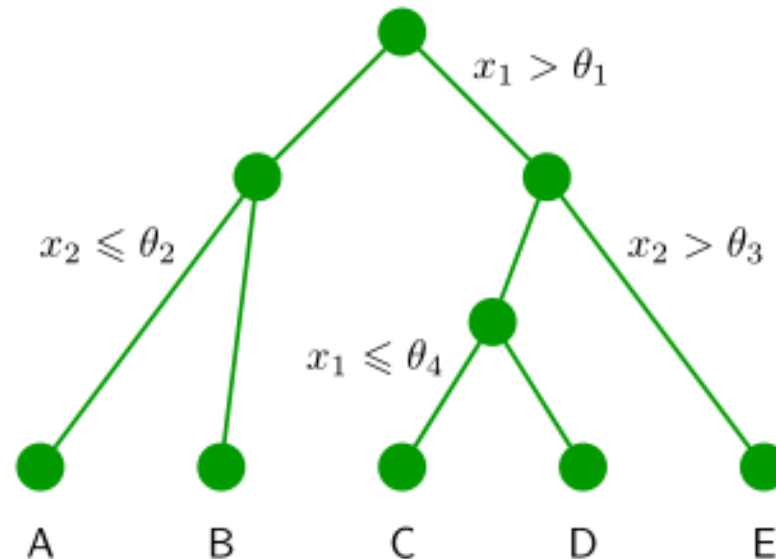
$$F : D_{x_1} \times D_{x_2} \times \dots \times D_{x_N} \rightarrow D_Y$$

- Recursively partitioning the input data space into non-overlapping regions
- Simple model each region
  - constant
  - simple function



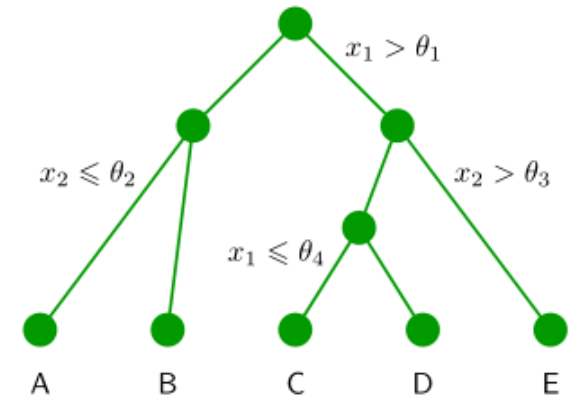
# Tree Model

- Easy to interpret; thus popular



# Learning Tree Model

- Greedy learning algorithm



Input: node  $n$ , training dataset  $D$

1) fully scan  $D$ , find the *best split*, by maximizing ‘purity’

$$|D| \times \text{Var}(D) - (|D_L| \times \text{Var}(D_L) + |D_R| \times \text{Var}(D_R))$$

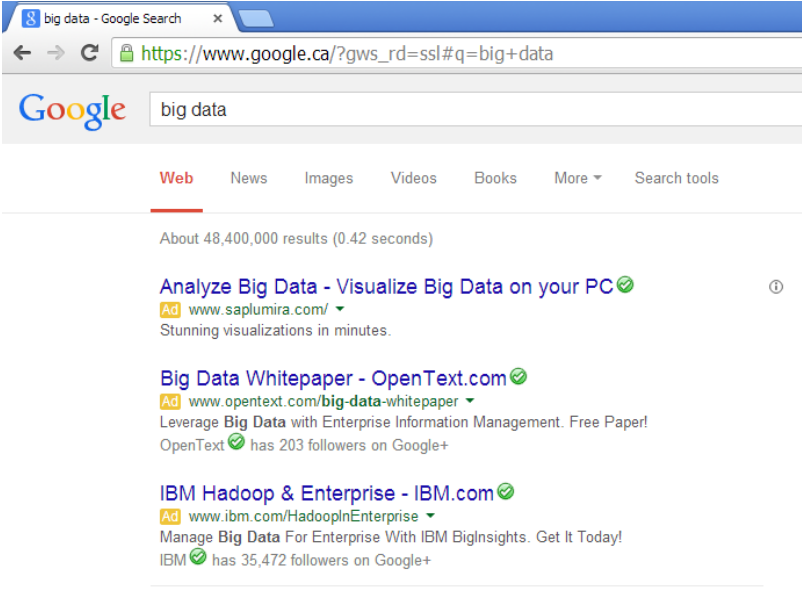
2) for either branch

- if stopping criteria satisfied: *pure* region
- else: advance a level

# Google's Bounce Rate Prediction Problem

- 'Bounce'
- High bounce rate = poor user experience
- Task: predicting bounce rate with data on hand

What if big data?



The screenshot shows a Google search for "big data" on the Canadian domain (google.ca). The search results page displays approximately 48,400,000 results in 0.42 seconds. The top results are advertisements:

- Analyze Big Data - Visualize Big Data on your PC** (Ad) from [www.saplumira.com/](http://www.saplumira.com/). Description: "Stunning visualizations in minutes."
- Big Data Whitepaper - OpenText.com** (Ad) from [www.opentext.com/big-data-whitepaper](http://www.opentext.com/big-data-whitepaper). Description: "Leverage Big Data with Enterprise Information Management. Free Paper! OpenText has 203 followers on Google+."
- IBM Hadoop & Enterprise - IBM.com** (Ad) from [www.ibm.com/HadoopInEnterprise](http://www.ibm.com/HadoopInEnterprise). Description: "Manage Big Data For Enterprise With IBM BigInsights. Get It Today! IBM has 35,472 followers on Google+."

# Learning Tree Model

- Greedy learning algorithm

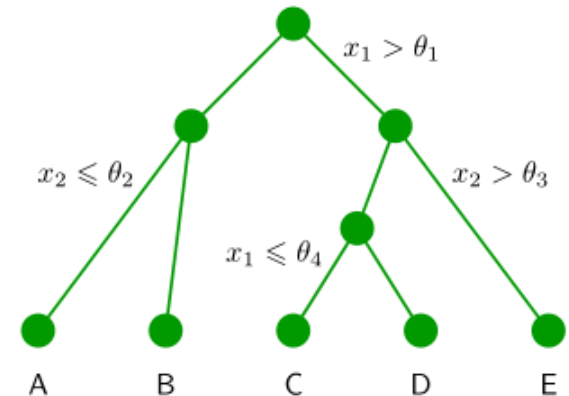
Input: node  $n$ , training dataset  $D$

1) **fully scan  $D$ , find the *best split***

- out of memory
- hard disk slow

2) for either branch

- if stopping criteria satisfied: *pure* region
- else: build a higher-level node



# Solution - Scaling Up Tree Learning

- Fully scan  $D$ , find the *best split* - out of memory  
- hard disk slow
- By Google Research, 2009
  - Computer Cluster
  - MapReduce
  - Tree learning

Panda, Biswanath, et al. "Planet: massively parallel learning of tree ensembles with mapreduce." *Proceedings of the VLDB Endowment* 2.2 (2009): 1426-1437.

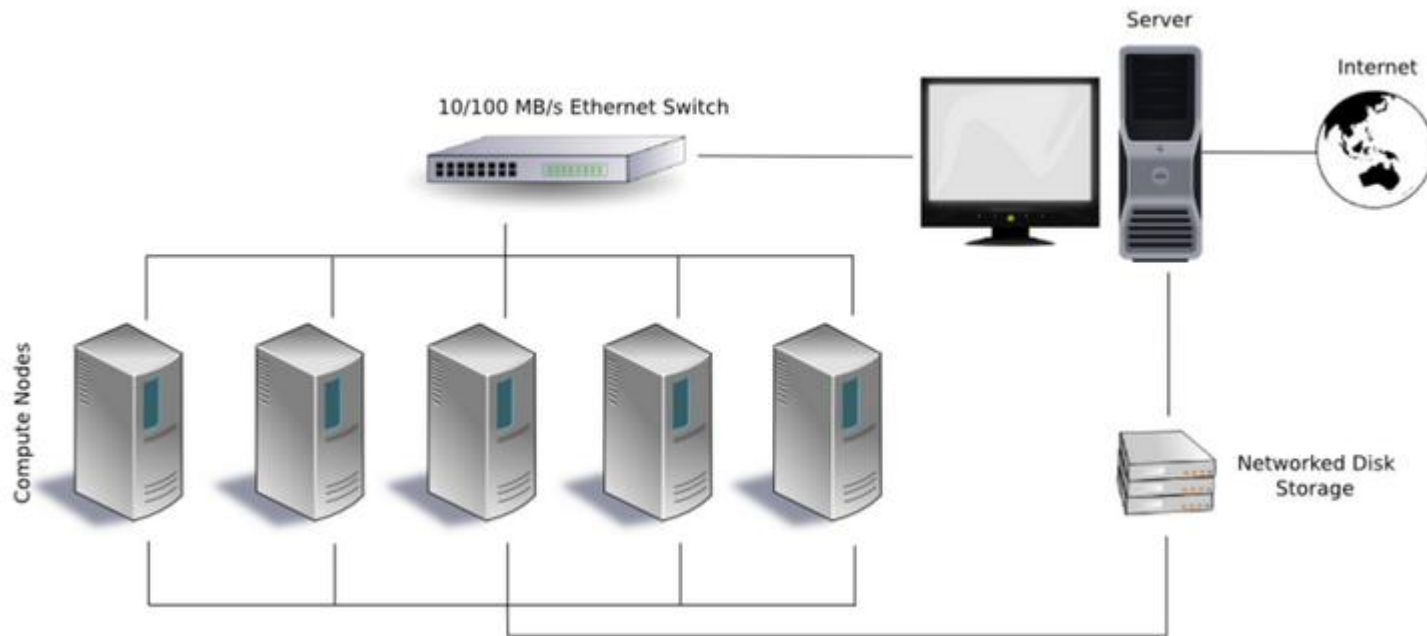
# Content

- Background
- **Methodology**
- Results
- Conclusion



# Computer Cluster

- Controller and workers



Source: Wikipedia



# MapReduce Framework

- Objective: to easily handle data too large to fit in memory

# MapReduce Framework

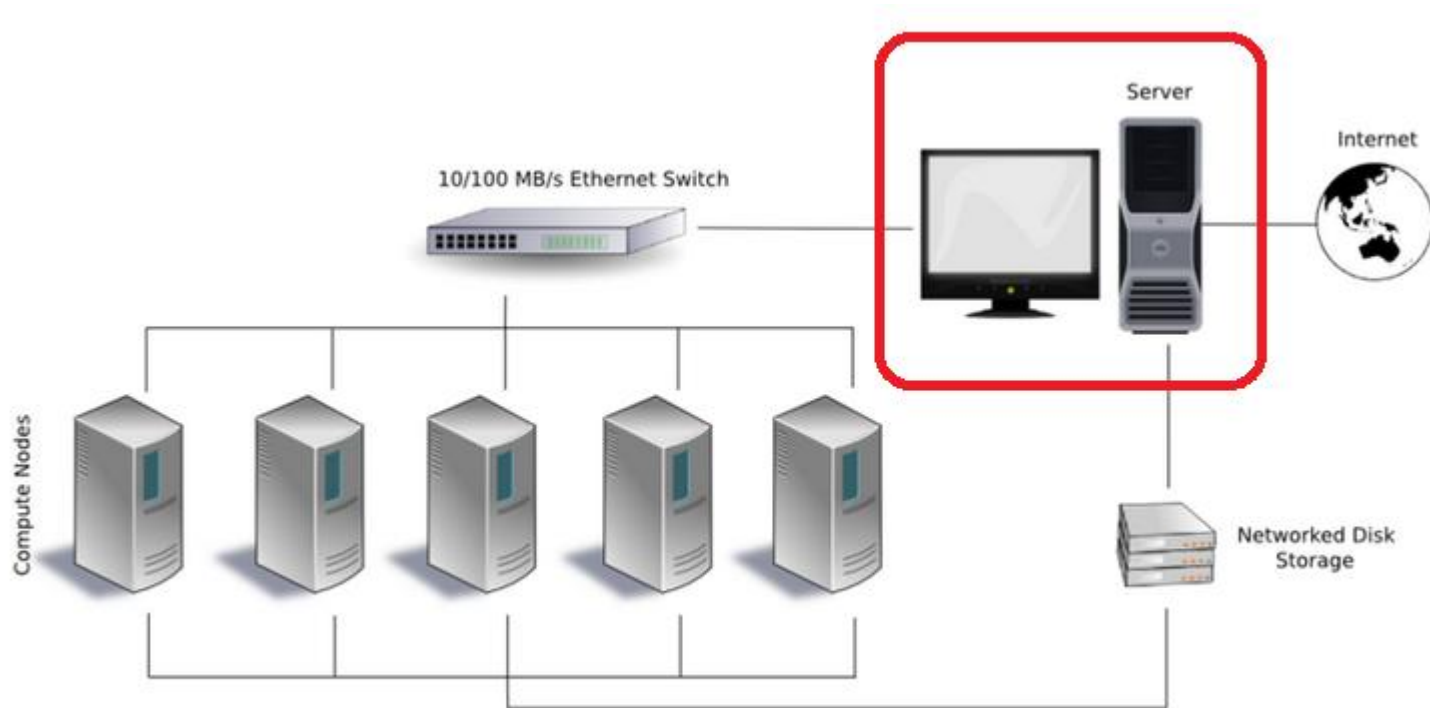
- Objective: to easily handle data too large to fit in memory
- It does all the dirty work:
  - distribute the data
  - parallelize the computation
  - handle failures

# MapReduce Framework

- Objective: to easily handle data too large to fit in memory
- It does all the dirty work:
  - distribute the data
  - parallelize the computation
  - handle failures
- User simply writes Map and Reduce functions

# Computer Cluster

- Core: Controller



Source: Wikipedia

# Job of Controller

- Keeps model file (M), containing the entire tree constructed so far
- Partitions the whole training dataset, across a set of mappers

# Job of Controller

- Each tree node, detects size of data set

if single machine ok?

-> push to 'SmallData Queue'

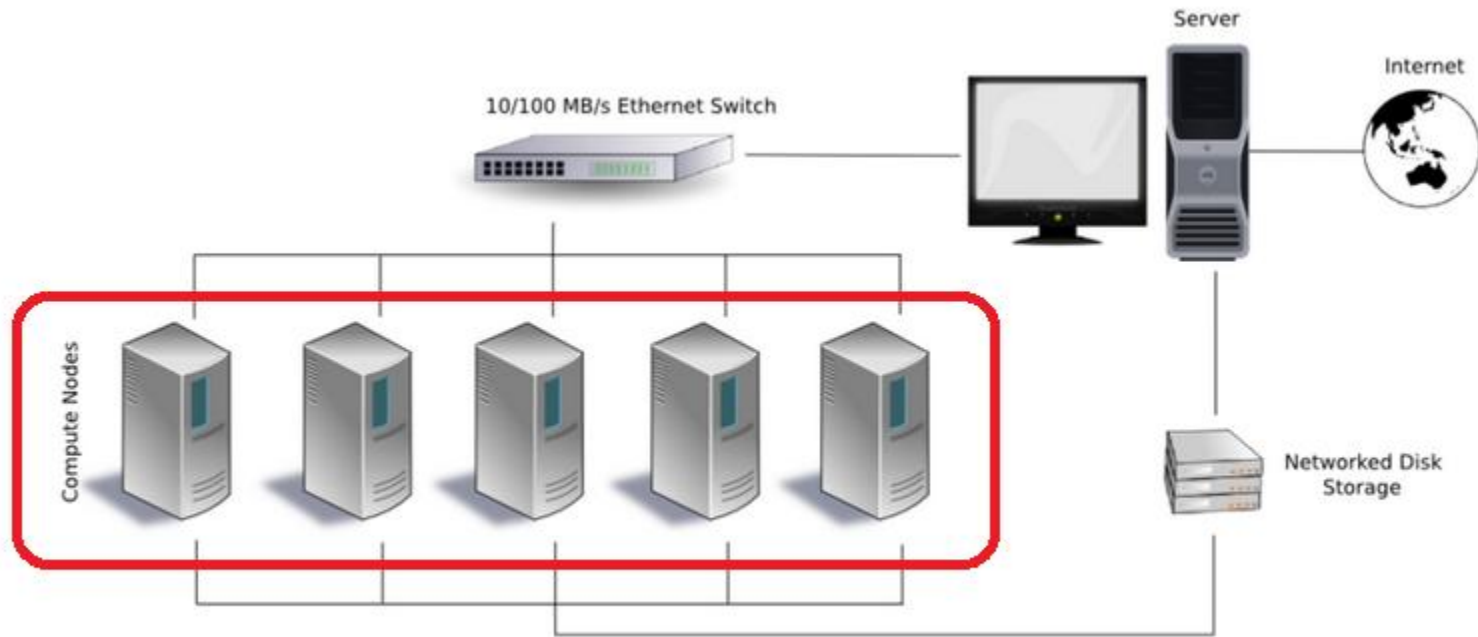
else

-> push to 'LargeData Queue'

- Schedules jobs in both queues for workers

# Job of Workers

- Map and Reduce functions



Source: Wikipedia

# MapReduce Work - SmallData Queue

- Map function
  - input:
    - partitioned training set  $D_k$
    - node  $n$
    - Model file M
  - check if an instance input to  $n$  -> emits
  - output (list):
    - key = node  $n$
    - value = subset of  $D_k$  input to  $n$



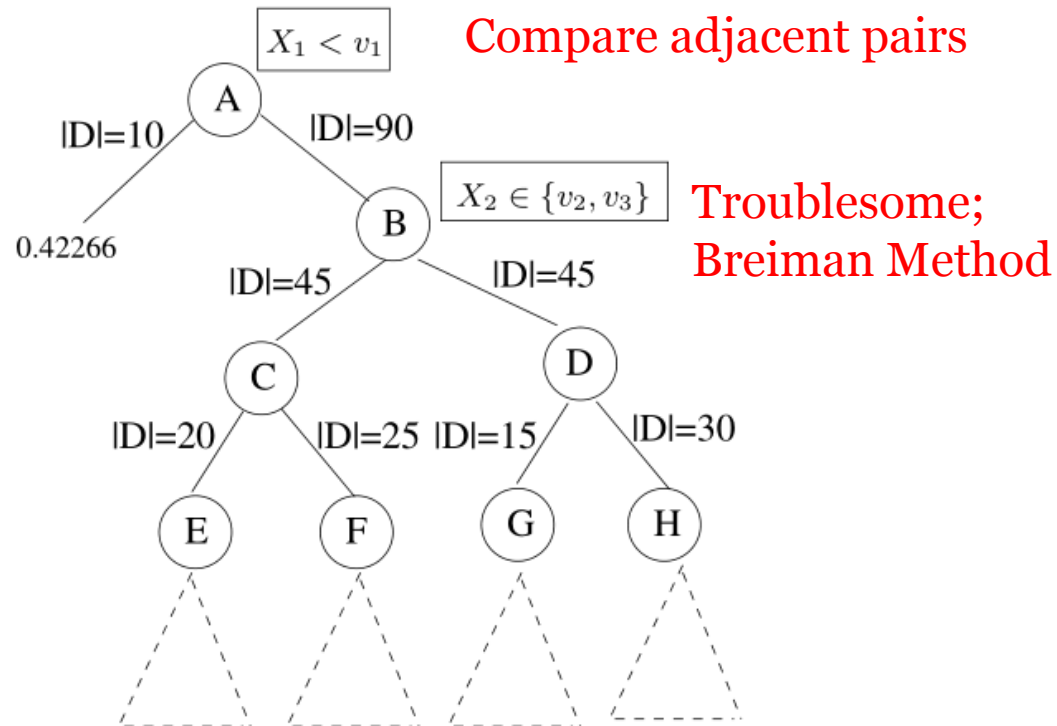
# MapReduce Work - SmallData Queue

- Reduce function
  - input:
    - key = node  $n$
    - value = subset of  $D_k$  input to  $n$
  - loads training records in memory
  - single-machine algorithm to find the *split*

In this way, cluster can process many nodes in parallel to grow the tree

# MapReduce Work – LargeData Queue

- Ordered attribute vs. Unordered



# MapReduce Work – LargeData Queue

- Map function

---

## Algorithm 2 MR\_ExpandNodes::Map

---

**Require:** NodeSet  $N$ , ModelFile  $M$ , Training record  $(\mathbf{x}, y) \in D^*$

- 1:  $n = \text{TraverseTree}(M, \mathbf{x})$
- 2: **if**  $n \in N$  **then**
- 3:    $\text{agg\_tup}_n \leftarrow y$
- 4:   **for all**  $X \in \mathcal{X}$  **do**
- 5:      $v = \text{Value on } X \text{ in } \mathbf{x}$
- 6:     **if**  $X$  is ordered **then**
- 7:       **for all** Split point  $s$  of  $X$  s.t.  $s < v$  **do**
- 8:          $T_{n,X}[s] \leftarrow y$
- 9:     **else**
- 10:       $T_{n,X}[v] \leftarrow y$

---



---

## Algorithm 3 MR\_ExpandNodes::Map\_Finalize

---

**Require:** NodeSet  $N$

- 1: **for all**  $n \in N$  **do**
- 2:   Output to all reducers( $n, \text{agg\_tup}_n$ )
- 3:   **for all**  $X \in \mathcal{X}$  **do**
- 4:     **if**  $X$  is ordered **then**
- 5:       **for all** Split point  $s$  of  $X$  **do**
- 6:         Output( $(n, X, s), T_{n,X}[s]$ )
- 7:     **else**
- 8:       **for all**  $v \in T_{n,X}$  **do**
- 9:         Output( $(n, X), (v, T_{n,X}[v])$ )

---

# MapReduce Work – LargeData Queue

- Reduce function

---

**Algorithm 4** MR\_ExpandNodes::Reduce

---

**Require:** Key  $k$ , Value Set  $V$

```

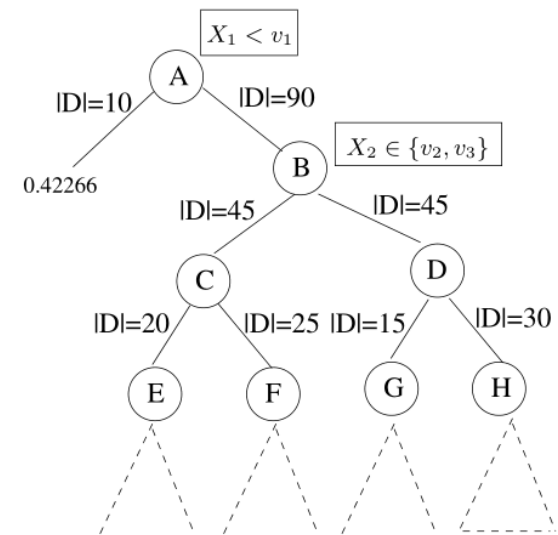
1: if  $k == n$  then
2:   {Aggregate  $\text{agg\_tup}_n$ 's from mappers}
3:    $\text{agg\_tup}_n = \text{Aggregate}(V)$ 
4: else if  $k == n, X, s$  then
5:   {Split on ordered attribute}
6:    $\text{agg\_tup}_{\text{left}} = \text{Aggregate}(V)$ 
7:    $\text{agg\_tup}_{\text{right}} = \text{agg\_tup}_n - \text{agg\_tup}_{\text{left}}$ 
8:    $\text{UpdateBestSplit}(S[n], X, s, \text{agg\_tup}_{\text{left}}, \text{agg\_tup}_{\text{right}})$ 
9: else if  $k == n, X$  then
10:  {Split on unordered attribute}
11:  for all  $v, \text{agg\_tup} \in V$  do
12:     $T[v] \leftarrow \text{agg\_tup}$ 
13:   $\text{UpdateBestSplit}(S[n], \text{BreimanSplit}(X, T, \text{agg\_tup}_n))$ 

```

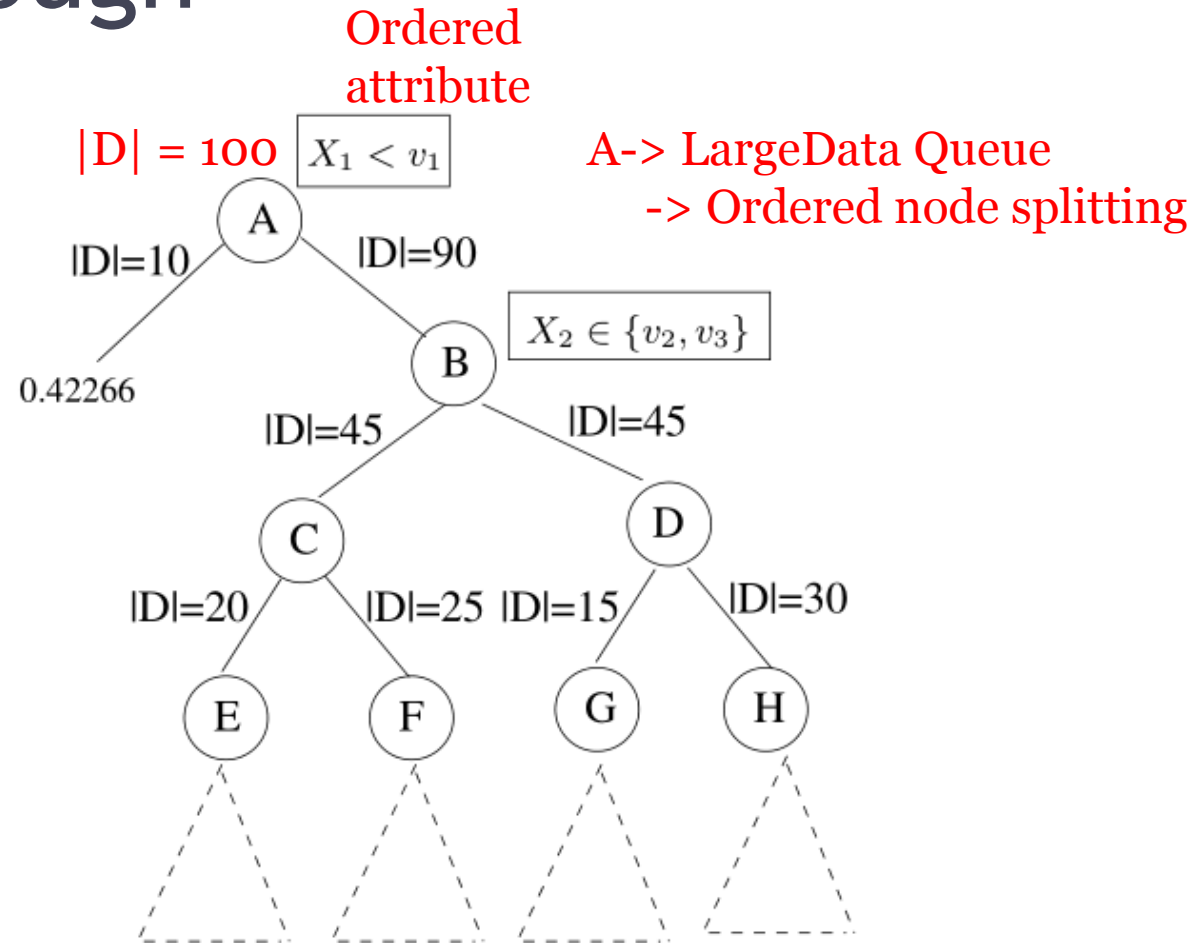
---

# Walkthrough

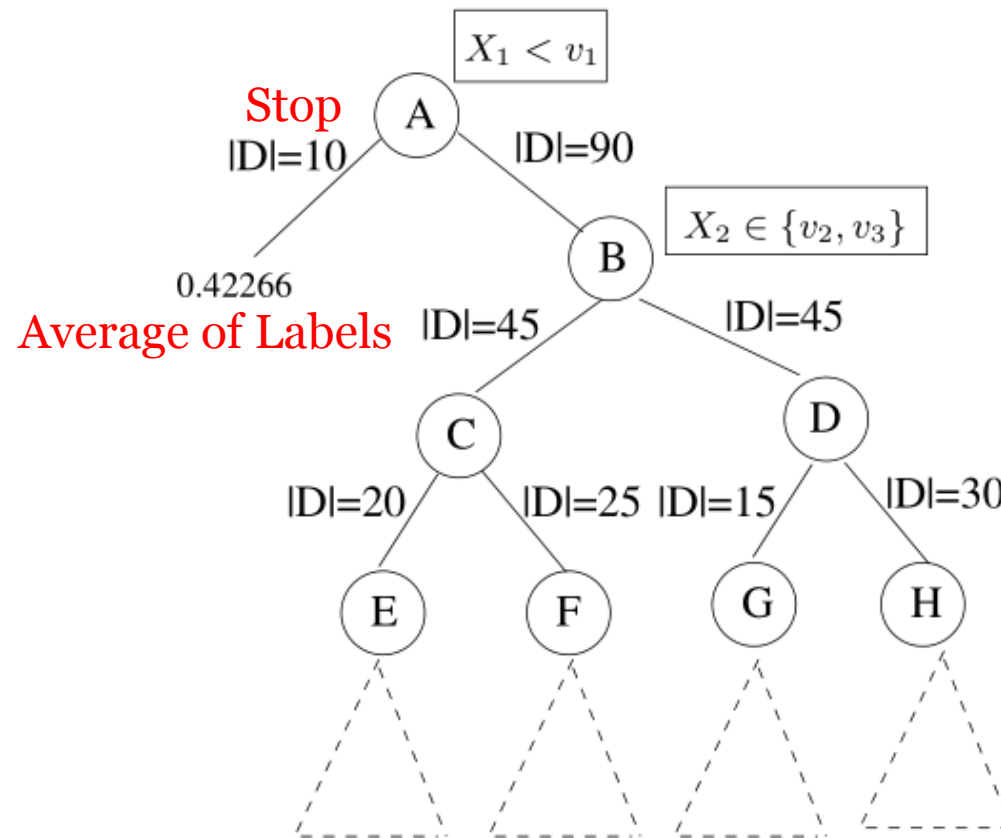
- Training set  $D^*$ 
  - 100 instances
- Memory constraint
  - 25 instances
- Stopping criteria
  - instances  $\leq 10$



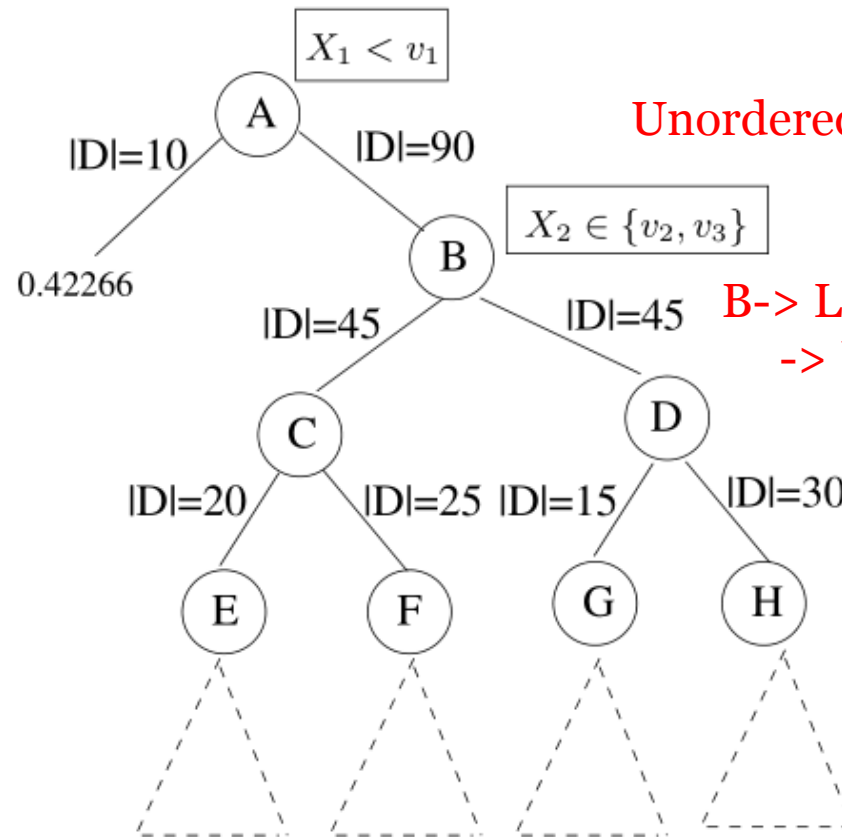
# Walkthrough



# Walkthrough

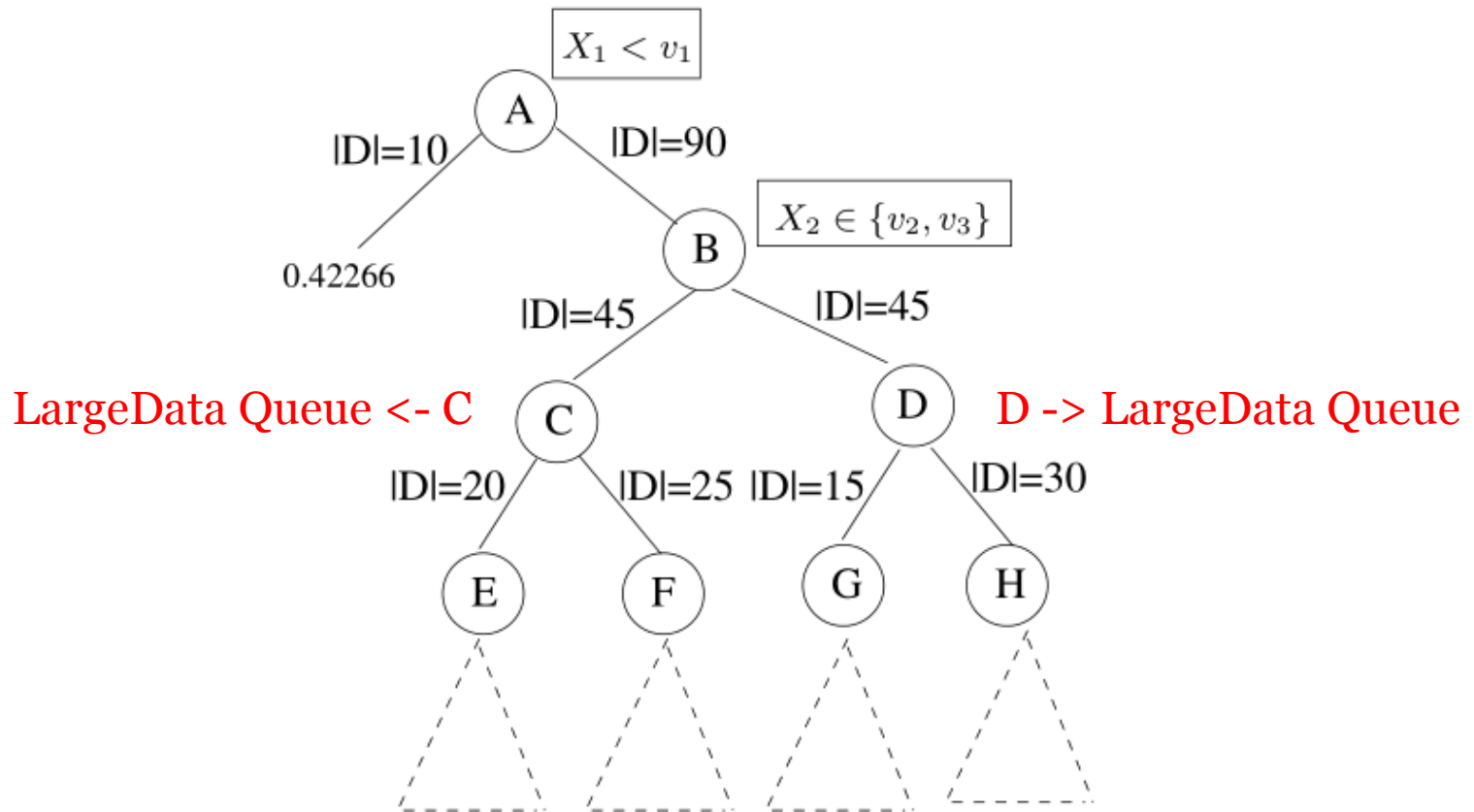


# Walkthrough

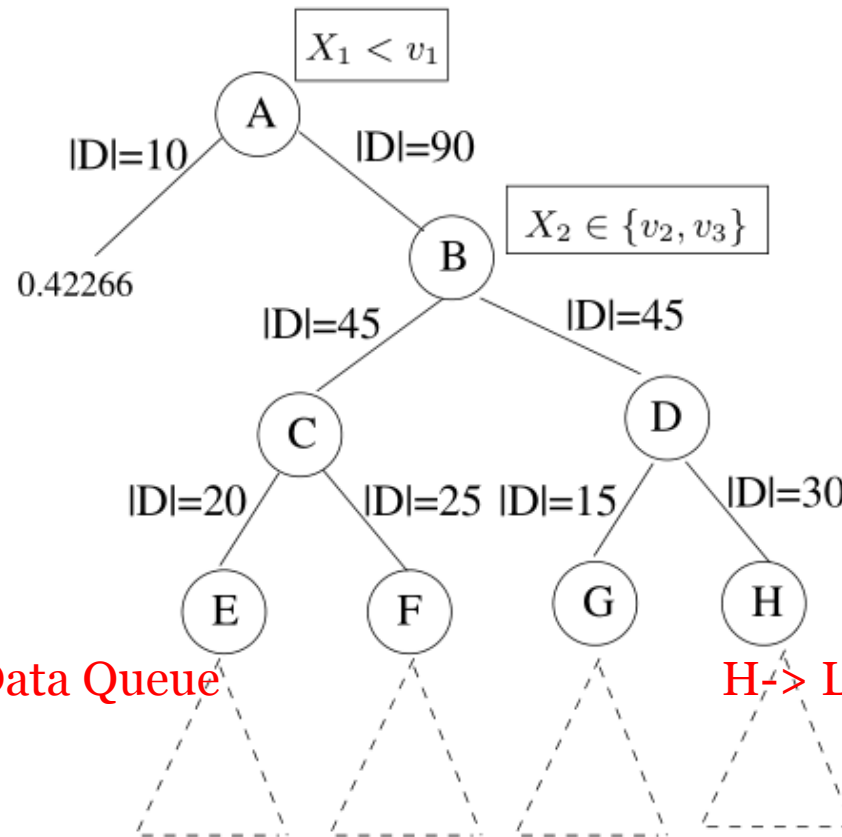




# Walkthrough



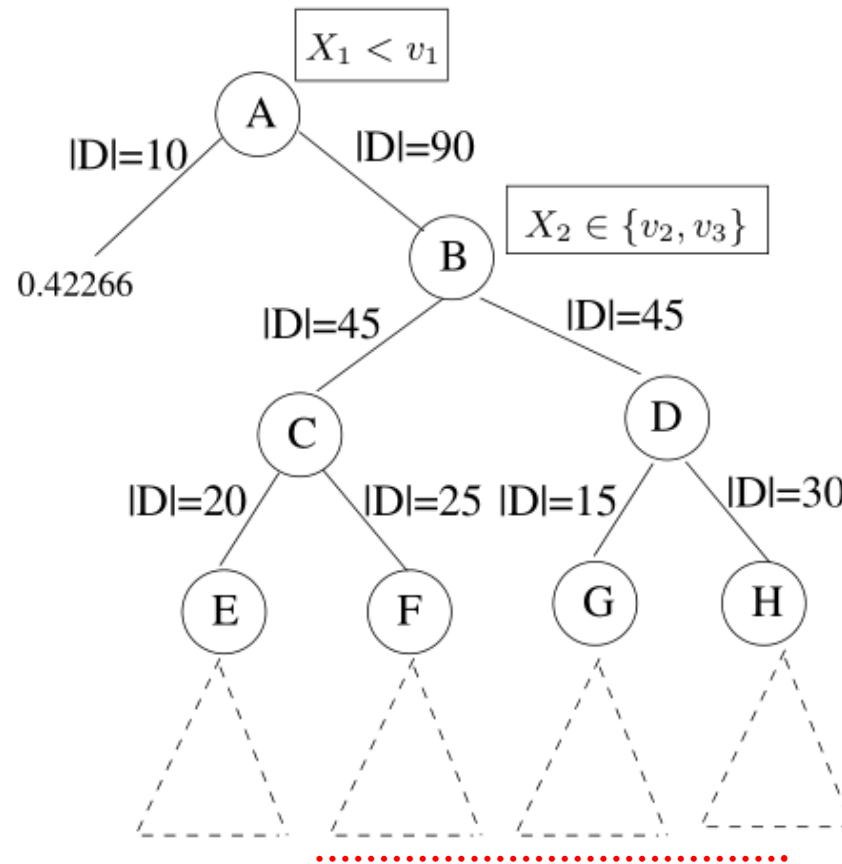
# Walkthrough



E, F, G -> SmallData Queue

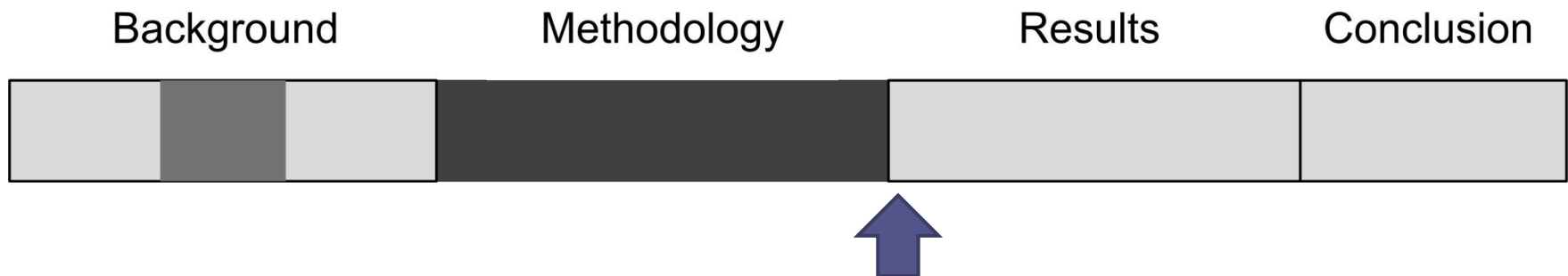
H-> LargeData Queue

# Walkthrough



# Content

- Background
- Methodology
- **Results**
- Conclusion

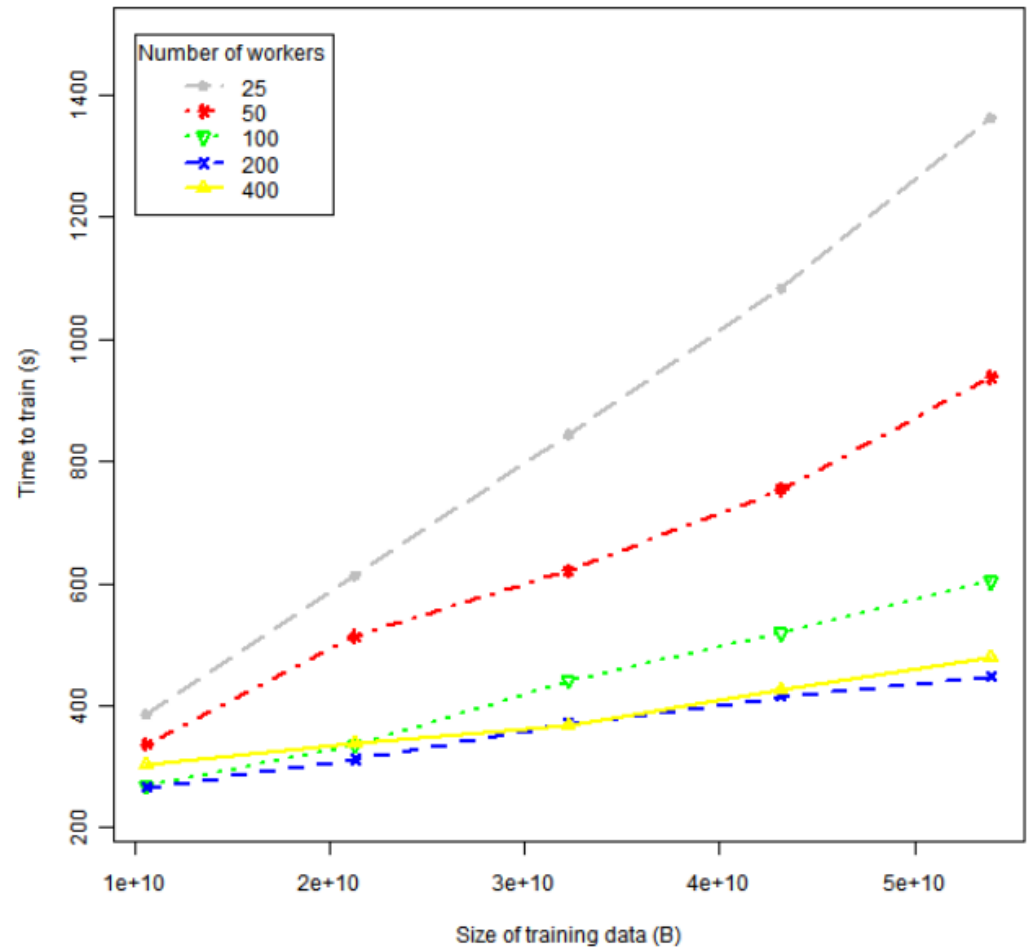


# Setup

- Bounce rate prediction problem
- 314 million records
  - 10 features
  - 1 label
- Each machine 768MB memory

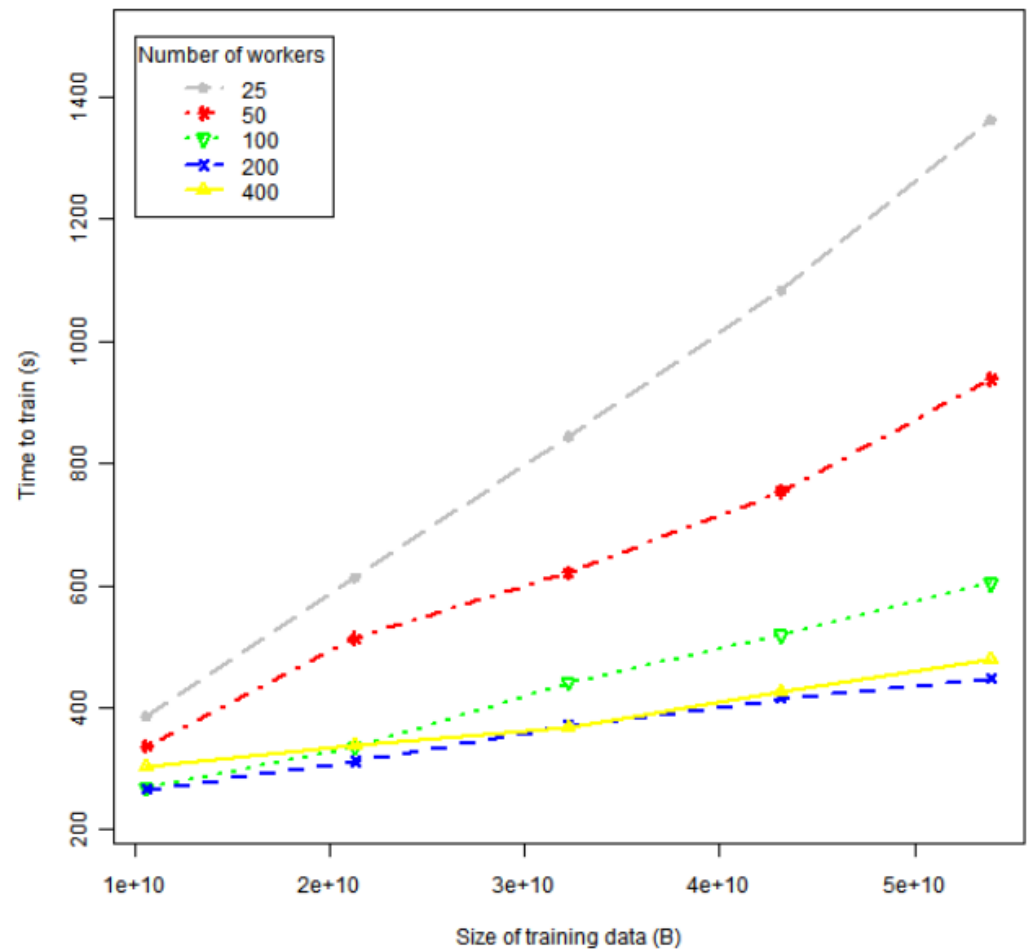
# Time to Train vs. Data Size

- Works well
  - 25 nodes



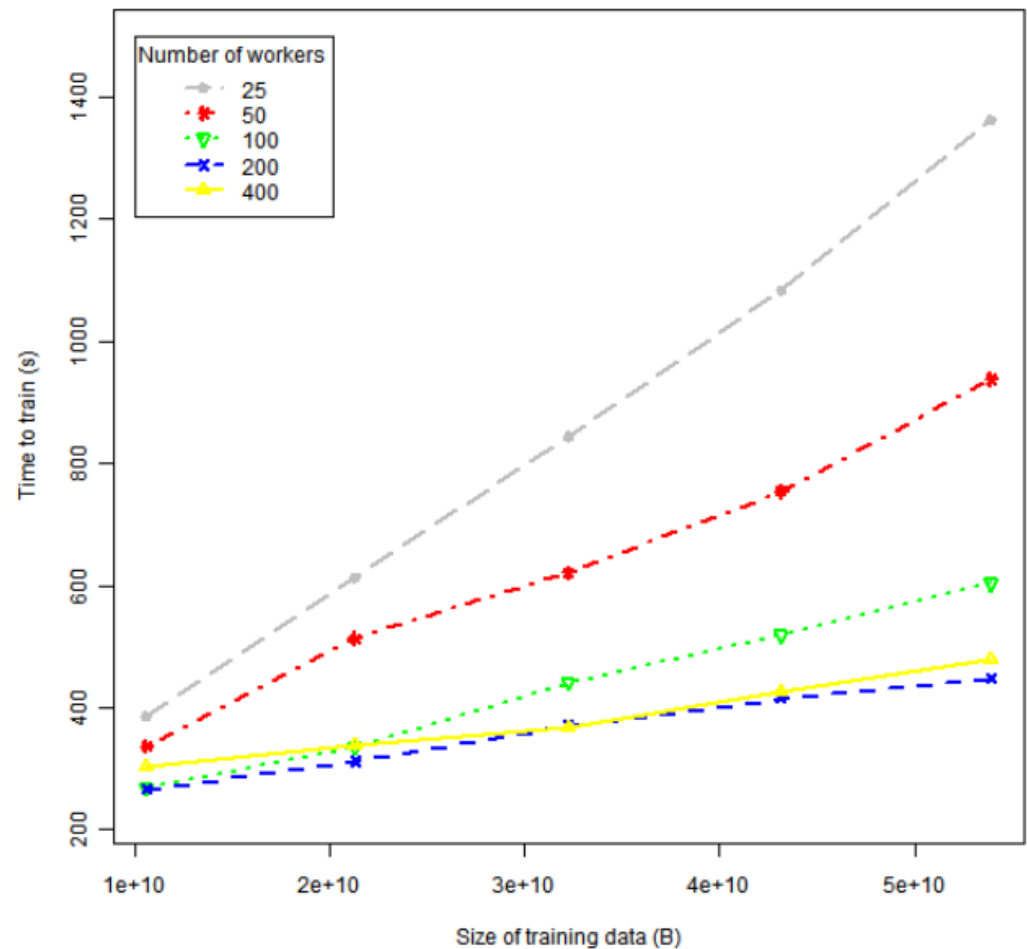
# Time to Train vs. Data Size

- Works well
  - 25 nodes
  - 50



# Time to Train vs. Data Size

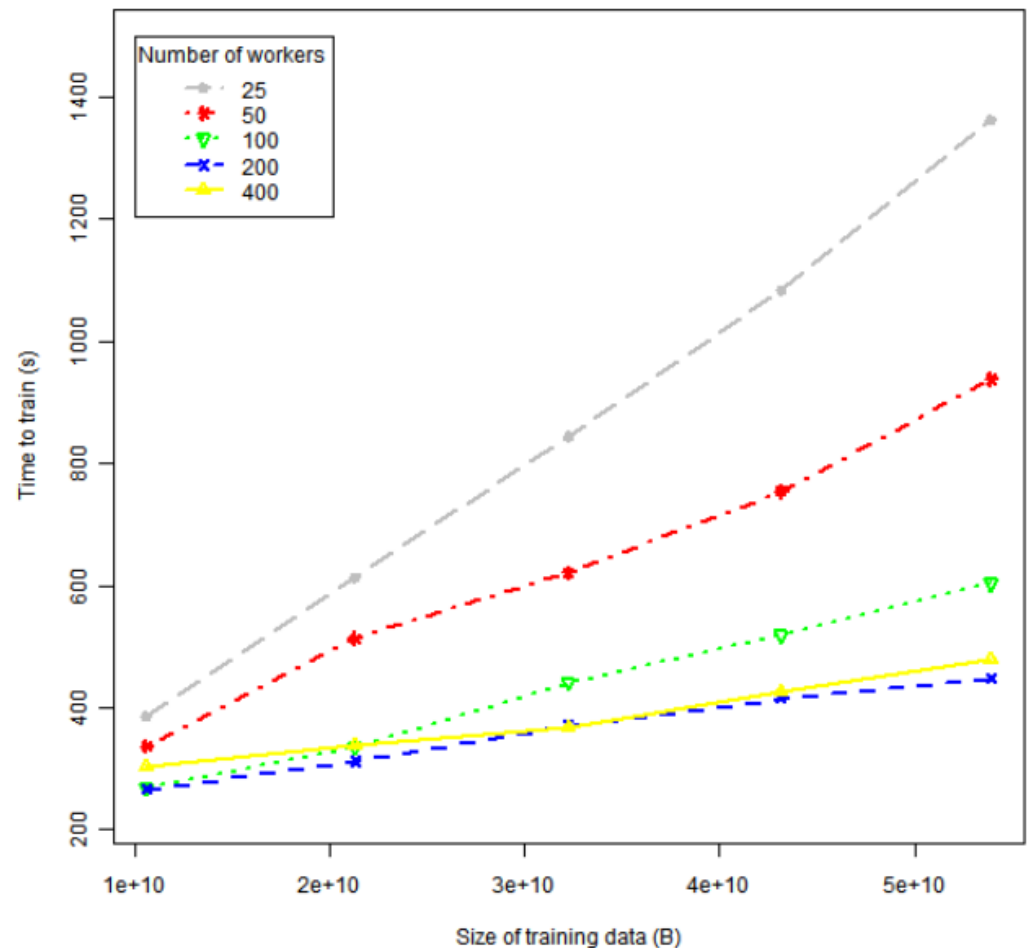
- Works well
  - 25 nodes
  - 50
  - 100
  - 200





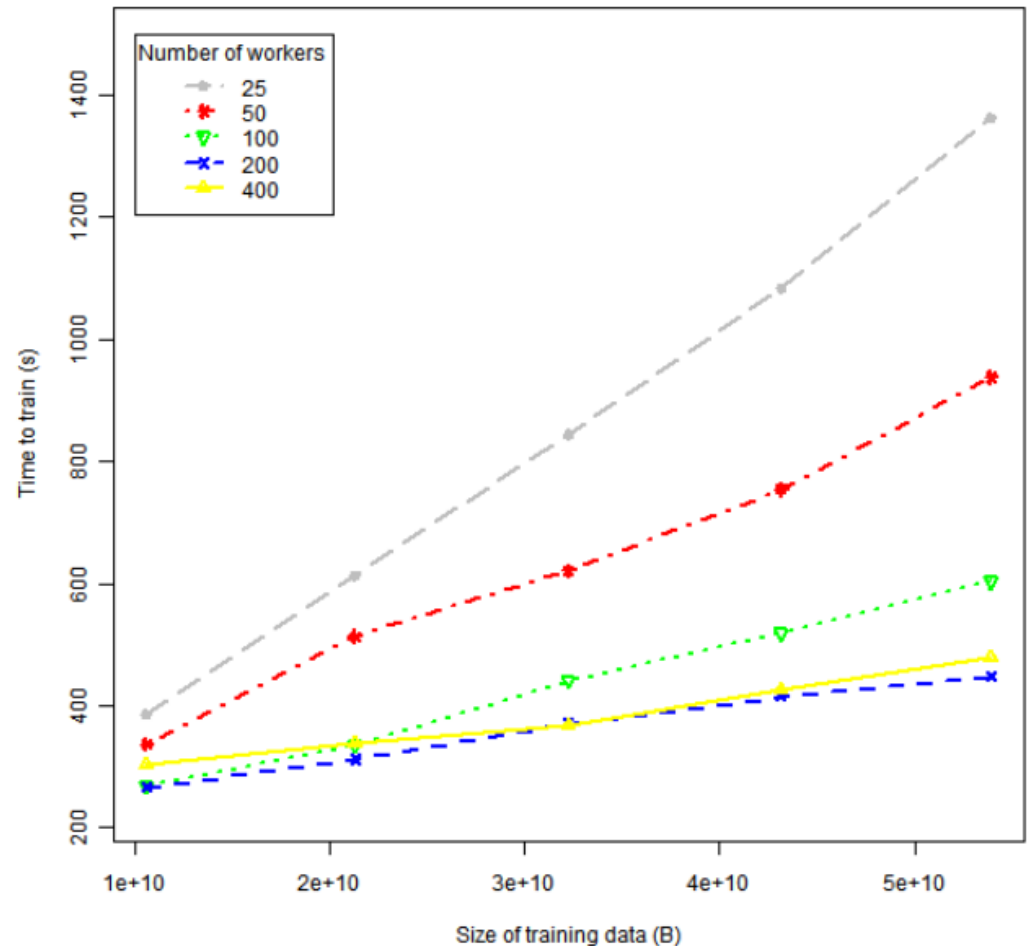
# Time to Train vs. Data Size

- Works well
  - 25 nodes
  - 50
  - 100
  - 200
  - 400?



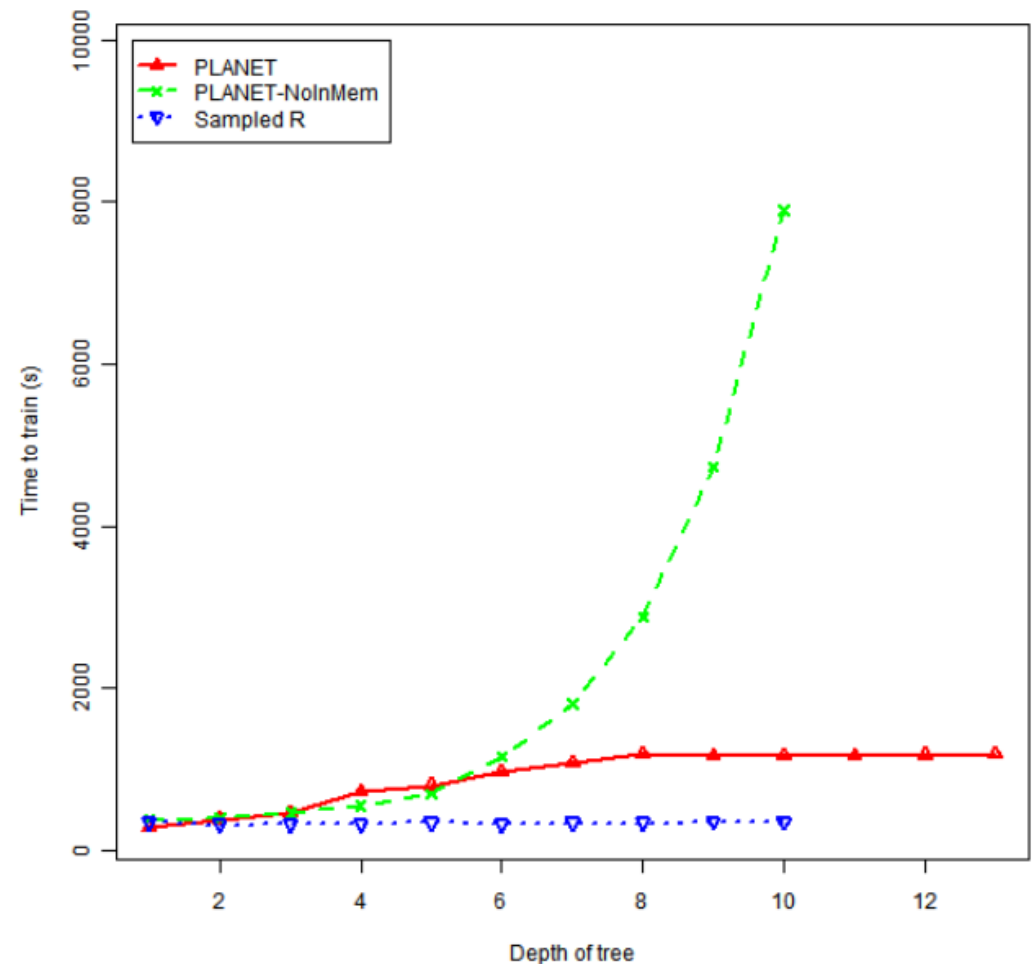
# Time to Train vs. Data Size

- 400 workers worse than 200?
- Cluster Management
  - network overhead
  - failure watching
  - schedule backups
  - data distribution & collection



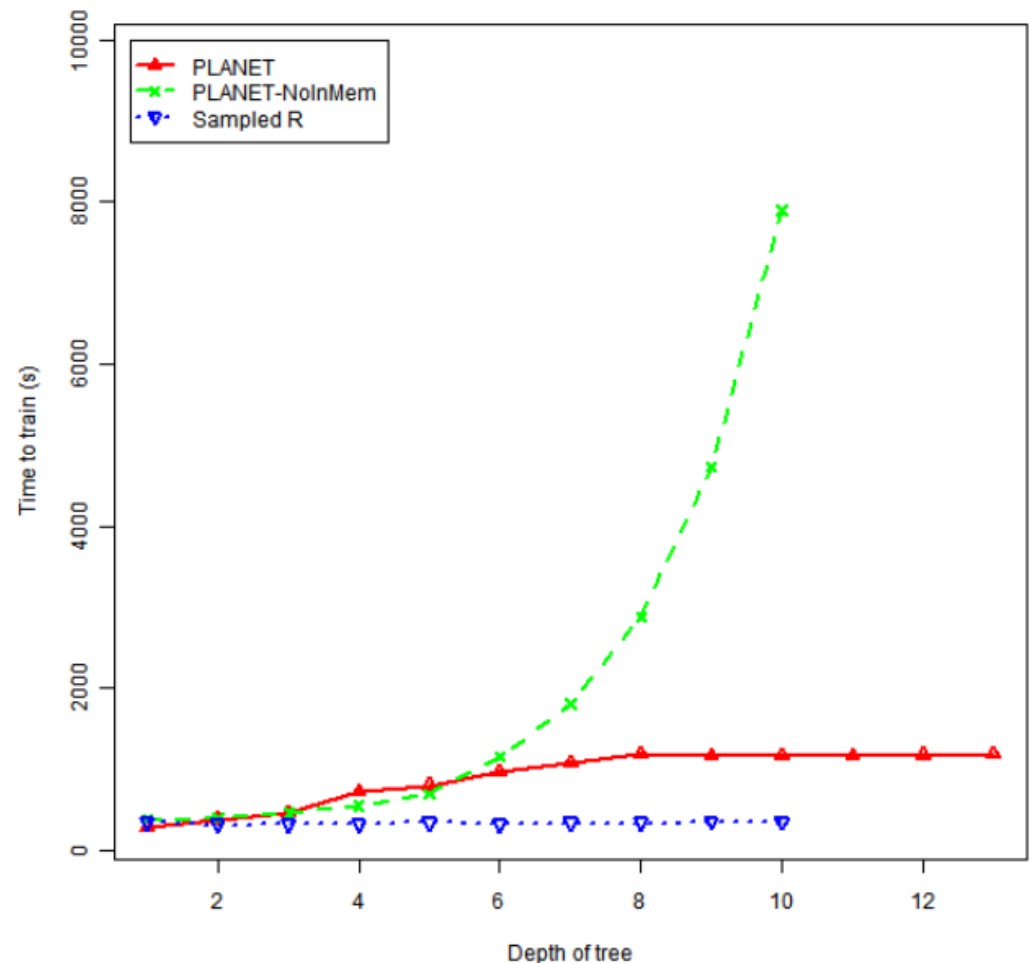
# Time to Train vs. Tree Depth

- With/without  
‘SmallData  
Queue’



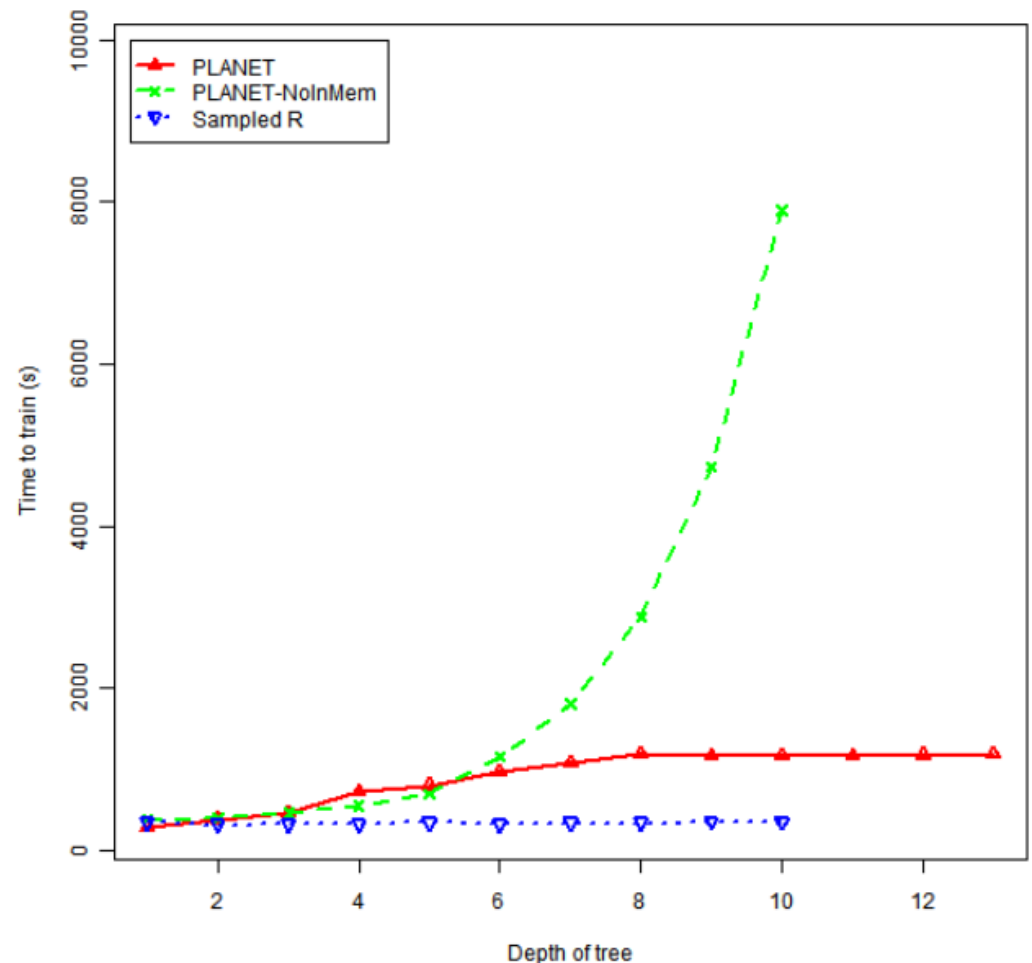
# Time to Train vs. Tree Depth

- With/without ‘SmallData Queue’
  - overhead of cluster management



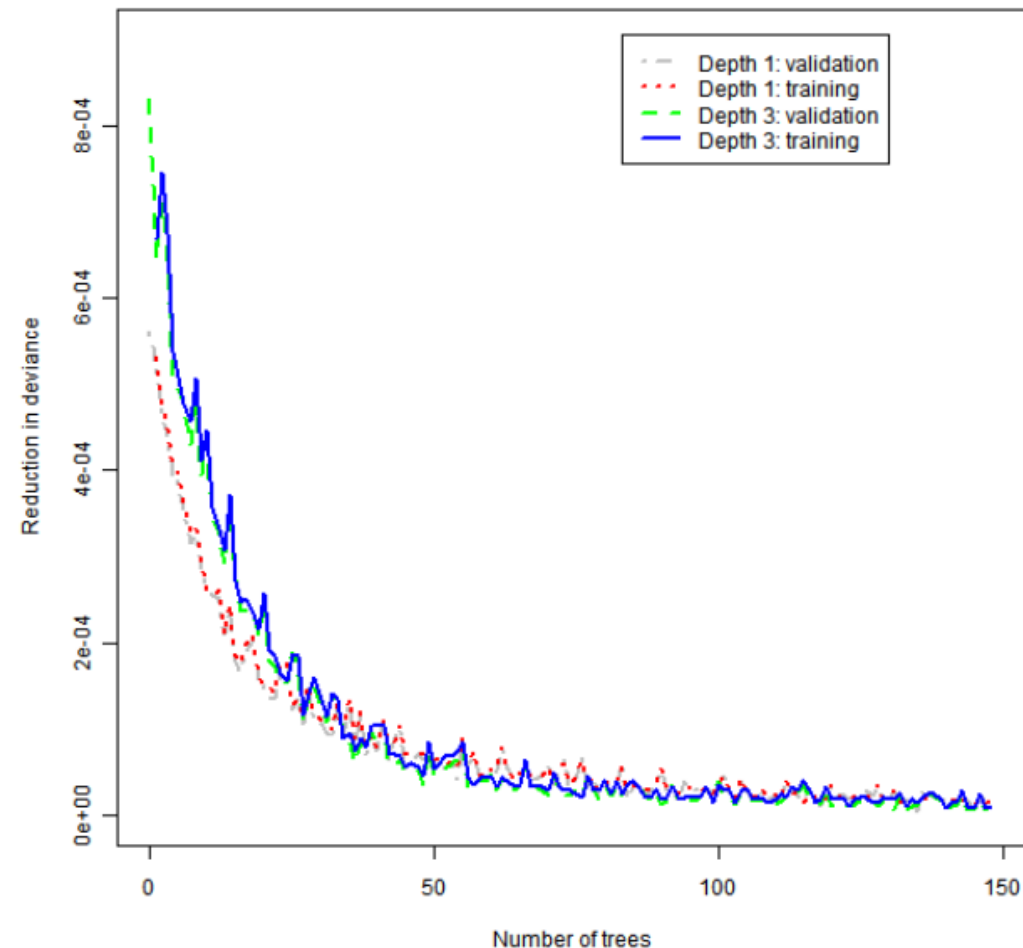
# Time to Train vs. Tree Depth

- With/without ‘SmallData Queue’
  - overhead of cluster management
  - sampling based method on single machine



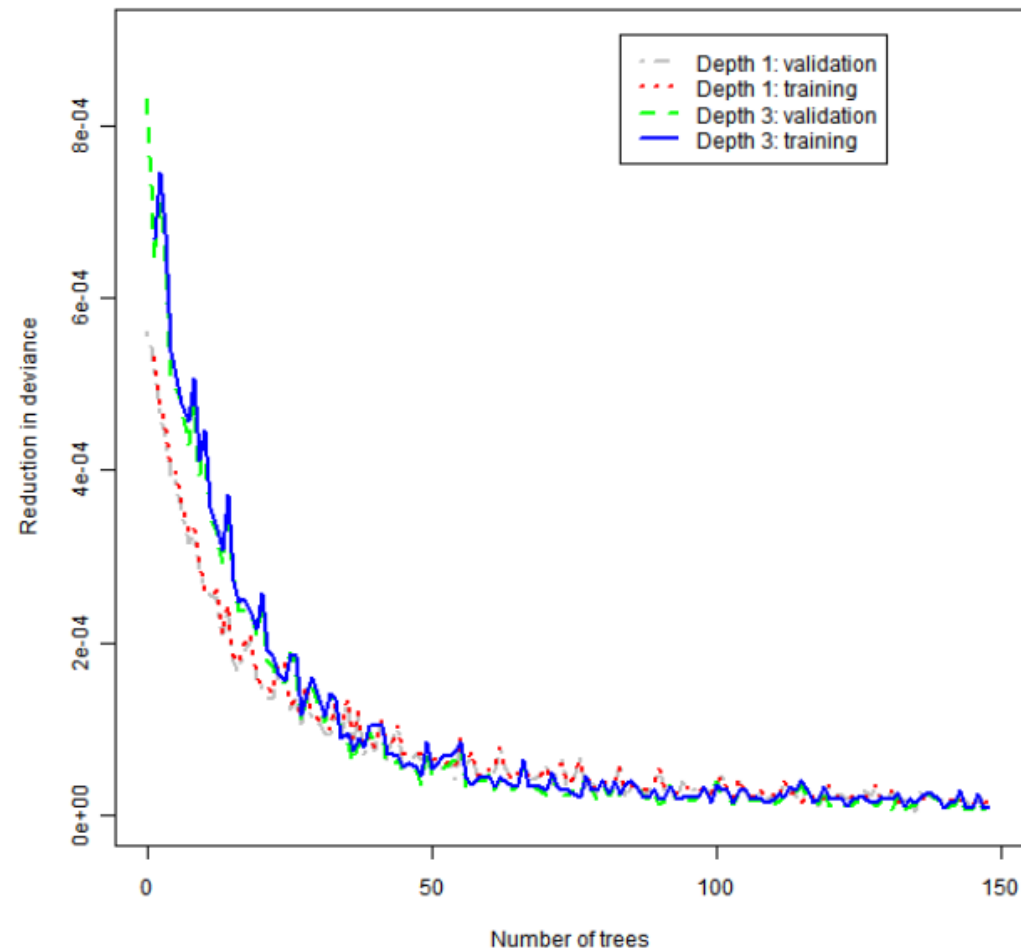
# Error Reduction vs Num of Trees

- Boosted tree model
  - a bundle of weighted weak learners: better performance



# Error Reduction vs Num of Trees

- Boosted tree model
  - a bundle of weighted weak learners: better performance
  - better weak learners faster error reduction



# Content

- Background
- Methodology
- Results
- **Conclusion**





# Conclusion

- Successfully scales up tree learning with MapReduce
- Performs well
- Pioneered large-scale machine learning

Bekkerman, Ron, Mikhail Bilenko, and John Langford, eds. *Scaling up machine learning: Parallel and distributed approaches*. Cambridge University Press, 2011.

## Related Work - Survey

Learning Setting	Algorithm	Cluster Nodes	Parallelization Framework	Speedup
Regression	Decision Tree	200	MapReduce	100
Classification	SVM	500	MPI	100
Ranking	LambdaMART	32	MPI	10
Inference	Loopy belief propagation	40	MPI	23
Inference	MCMC	1024	MPI	1000
Clustering	Spectral clustering	256	MapReduce, MPI	256
Clustering	Information-theoretic clustering	400	MPI	100

Go on. I dare you.



Source:  
Linkedin  
sharing

# References

1. Bishop, Christopher M. *Pattern recognition and machine learning*. Vol. 1. New York: springer, 2006.
2. Panda, Biswanath, et al. "Planet: massively parallel learning of tree ensembles with mapreduce." *Proceedings of the VLDB Endowment* 2.2 (2009): 1426-1437.
3. Dean, Jeffrey, and Sanjay Ghemawat. "MapReduce: simplified data processing on large clusters." *Communications of the ACM* 51.1 (2008): 107-113.
4. Bekkerman, Ron, Mikhail Bilenko, and John Langford, eds. *Scaling up machine learning: Parallel and distributed approaches*. Cambridge University Press, 2011.
5. Garcia-Molina, Hector. *Database systems: the complete book*. Pearson Education India, 2008.