Building Decision Trees Using MapReduce

Mining of Massive Datasets Leskovec, Rajaraman, and Ullman Stanford University



Problem: Building a tree

- Given a large dataset with hundreds of attributes
- Build a decision tree!
- General considerations:
 - Tree is small (can keep it memory):
 - Shallow (~10 levels)
 - Dataset too large to keep in memory
 - Dataset too big to scan over on a single machine
 - MapReduce to the rescue!

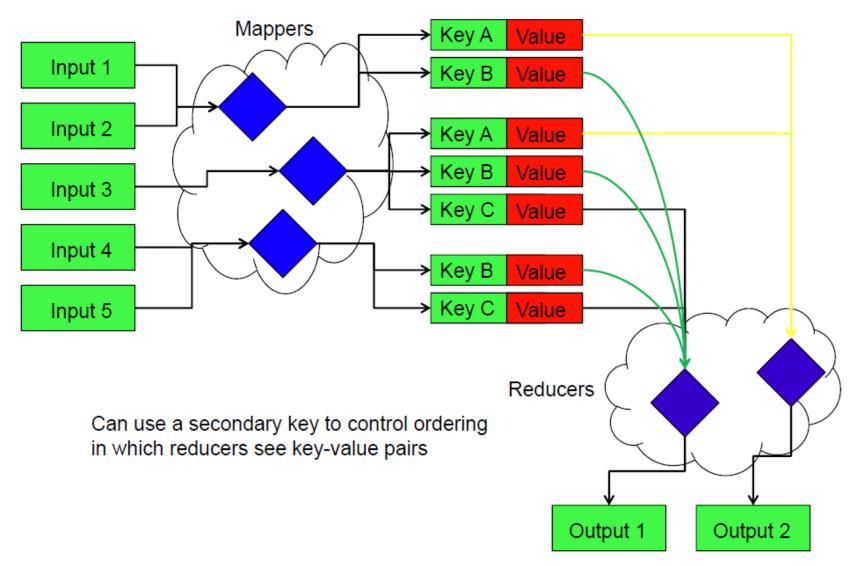
```
Algorithm 1 BuildSubTree

Require: Node n, Data D \subseteq D^*
1: (n \to \text{split}, D_L, D_R) = \text{FindBestSplit}(D)
2: if StoppingCriteria(D_L) then
3: n \to \text{left\_prediction} = \text{FindPrediction}(D_L)
4: else
5: BuildSubTree(n \to \text{left}, D_L)
```

- 7: $n \rightarrow \text{right_prediction} = \text{FindPrediction}(D_R)$ 8: else
- 9: **BuildSubTree** $(n \rightarrow \text{right}, D_R)$

6: if StoppingCriteria(D_R) then

MapReduce



Today's Lecture: PLANET

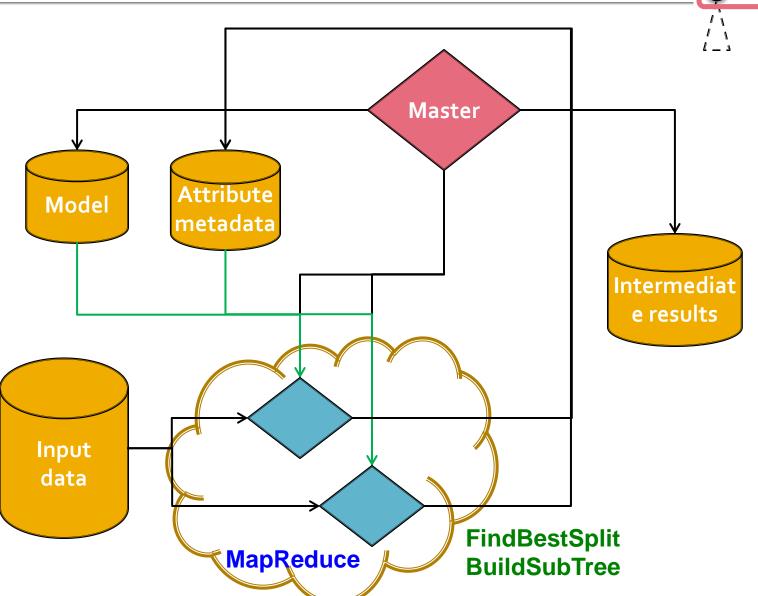
Parallel Learner for Assembling Numerous Ensemble Trees [Panda et al., VLDB '09]

 A sequence of MapReduce jobs that builds a decision tree

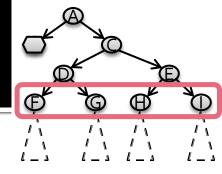
Setting:

- Hundreds of numerical (discrete & continuous, but not categorical) attributes
- Target variable is numerical: Regression
- Splits are binary: X_j < v</p>
- Decision tree is small enough for each Mapper to keep it in memory
- Data too large to keep in memory

PLANET Architecture

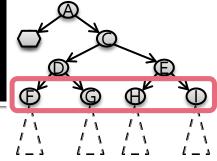


PLANET Overview



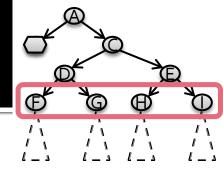
- We build the tree level by level
 - One MapReduce step builds one level of the tree
- Mapper
 - Considers a number of possible splits (X_i,v) on its subset of the data
 - For each split it stores partial statistics
 - Partial split-statistics is sent to Reducers
- Reducer
 - Collects all partial statistics and determines best split
- Master grows the tree for one level

PLANET Overview



- Mapper loads the model and info about which attribute splits to consider
 - Each mapper sees a subset of the data D*
 - Mapper "drops" each datapoint to find the appropriate leaf node L
 - For each leaf node L it keeps statistics about
 - (1) the data reaching L
 - (2) the data in left/right subtree under split S
- Reducer aggregates the statistics (1), (2) and determines the best split for each tree node

PLANET: Components



- Master
 - Monitors everything (runs multiple MapReduce jobs)
- Three types of MapReduce jobs:
 - (1) MapReduce <u>Initialization</u> (run once first)
 - For each attribute identify values to be considered for splits
- (2) MapReduce FindBestSplit (run multiple times)
 - MapReduce job to find best split when there is too much data to fit in memory
- (3) MapReduce <u>InMemoryBuild</u> (run once last)
 - Similar to BuildSubTree (but for small data)
 - Grows an entire sub-tree once the data fits in memory
- Model file
 - A file describing the state of the model

References

- B. Panda, J. S. Herbach, S. Basu, and R. J. Bayardo.
 PLANET: Massively parallel learning of tree ensembles with MapReduce. In Proc. VLDB 2009.
- J. Ye, J.-H. Chow, J. Chen, Z. Zheng. Stochastic Gradient Boosted Distributed Decision Trees. In Proc. CIKM 2009.