Distributed Random Forest Implementation

Sumeet Mishra
sumish@iu.edu
Indiana University Bloomington

Srinithish K
skandag@iu.edu
Indiana University Bloomington

KEYWORDS

airline dataset, random forest, spark, tensorflow

1 DATASET USED

We intend to use the Airline dataset

2 TASKS

(1) Install and configure Spark and Tensorflow on FutureSystem Nodes.
(2) Implement task parallel approach of random forest on Spark
(3) Combine task parallel with data parallel approach of random forest on Spark.
(4) Implement task parallel approach on TensorFlow.
(5) Compare and analyze the performances with different configuration settings.

3 PROJECT IDEA

Firstly, We intend to build a distributed the Random Forest algorithm on Spark framework. Random Forest has a natural suitability for parallelism given that the algorithm is a ensemble learning.

We would use the Airlines Data set to train and test the models performance

3.1 Spark

- We intend to use PySpark for easier interface with the Spark environment.For this we would have to install Spark on the FutureSystems Nodes.
- We want to implement 'Parallel Random Forest' as suggested in the paper [3] where the optimization is first achieved by Task Parallel approach, where decision trees are grown in a parallel manner.
- Second approach is to achieve optimization using Data Parallel approach where we would be vertically partitioning the data on each feature and save it in a Resilient Distributed Datasets (RDD) and hence the best split computation can occur close to the data
- Combining these two techniques we intend to build a robust algorithm that could scale to large data and achieve good accuracy.

3.2 Tensorflow

- We would first install and configure distributed tensorflow on the Future System nodes and interface with python.
- Implement task parallel approach [3] in the tensorflow and compare the running time of the forest with the equivalent in Spark.

We would use the 2008 data set and once we get the satisfied results we would combine 2008 and 2007 and compare the decrease in the performance if any in terms of computational time.

We would like to compare,

- Accuracy with number of decision trees grown.
- Compute time against the number of trees in the forest.
- The number of mappers and the compute time (Spark).

4 OPEN SOURCE CODE USED

Pyspark, Tensorflow, Scikit-learn.

5 PAPERS TO READ

. 1. Random Forests for BigData. Genuer,R.,Poggi,J.-M., Tuleau-Malot, C.,Villa-Vialaneix N.,2015.

2. A fast implementation of random forests for high dimensional data in C++ and R. Wright, M.N.,Ziegler, A.,2015.ranger.

3. A Parallel Random Forest Algorithm for Big Data in a Spark Cloud Computing Environment Jianguo Chen, Kenli Li et-al

6 MILESTONES

(1) Build distributed Random Forest on Spark in Task Parallel approach
(2) Implement Data Parallel approach on Spark
(3) Combine the two implementations on Spark.
(4) Task parallel approach on Tensorflow.
(5) If time permits we would want to also compare the performance of the implementation in Harp.

7 PORTION OF THE PROJECT AND RESPONSIBILITY

1. Random Forest Implementation: Sumeet and Nithish
2. Spark Implementation: Nithish
3. Tensorflow Implementation: Sumeet

, ,