Bag of Little Random Forests (BLRF)

Adapting the RF to a Big Data Setting

Motivation and Objectives

The Random Forests (RF) algorithm (Figure 1) is inefficient when handling Big Data:

- **Time**: significant time consumption;
- **Memory**: physical storage of big data sets;
- **Structure**: not well-adapted to a parallelism;
- **Unable** to load the entire data set into memory.

Aim to: build BLRF (Figure 3). Reduce computation time while maintaining prediction accuracy.

Steps and Procedures

- Study BLB and RF;
- Combine BLB and RF -> BLRF algorithm;
- Modify the source C code (regression only);
- Evaluate the BLRF algorithm (time and accuracy).

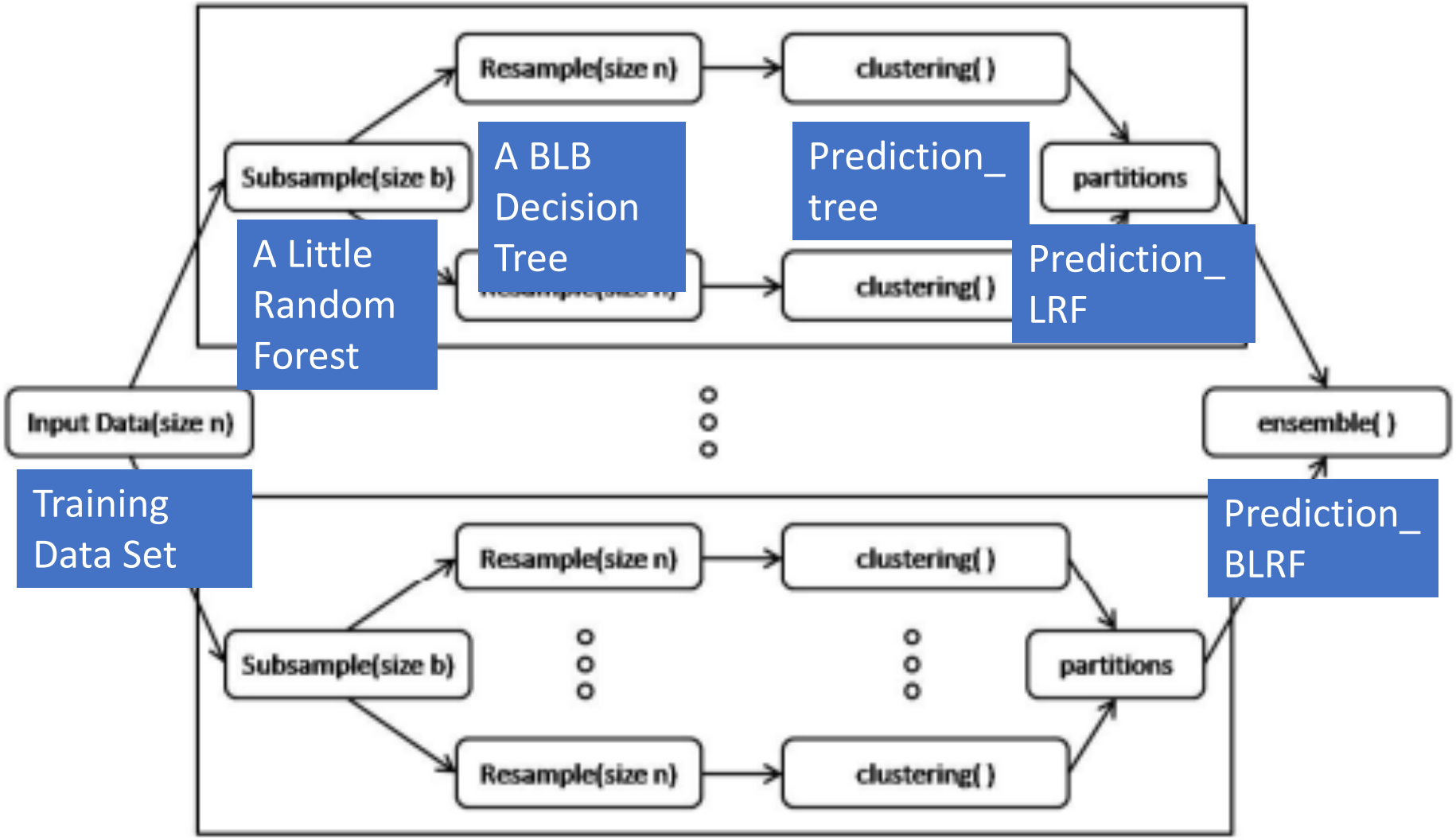


Figure 3. Visualization of BLRF

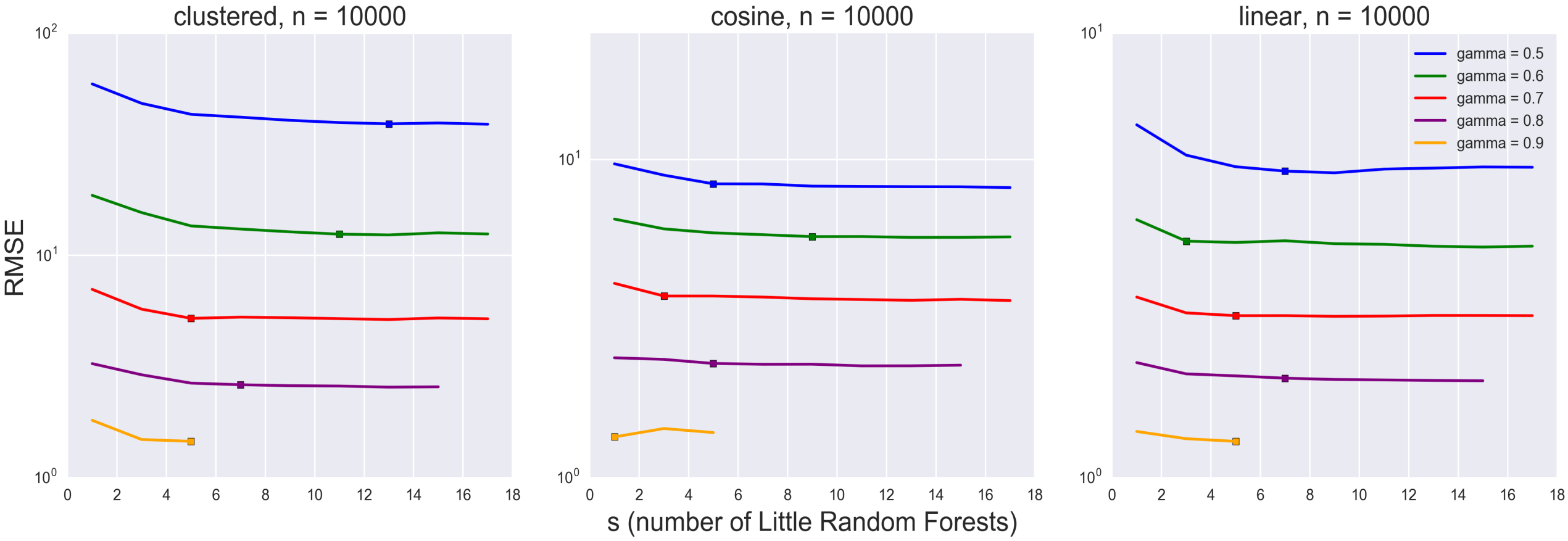
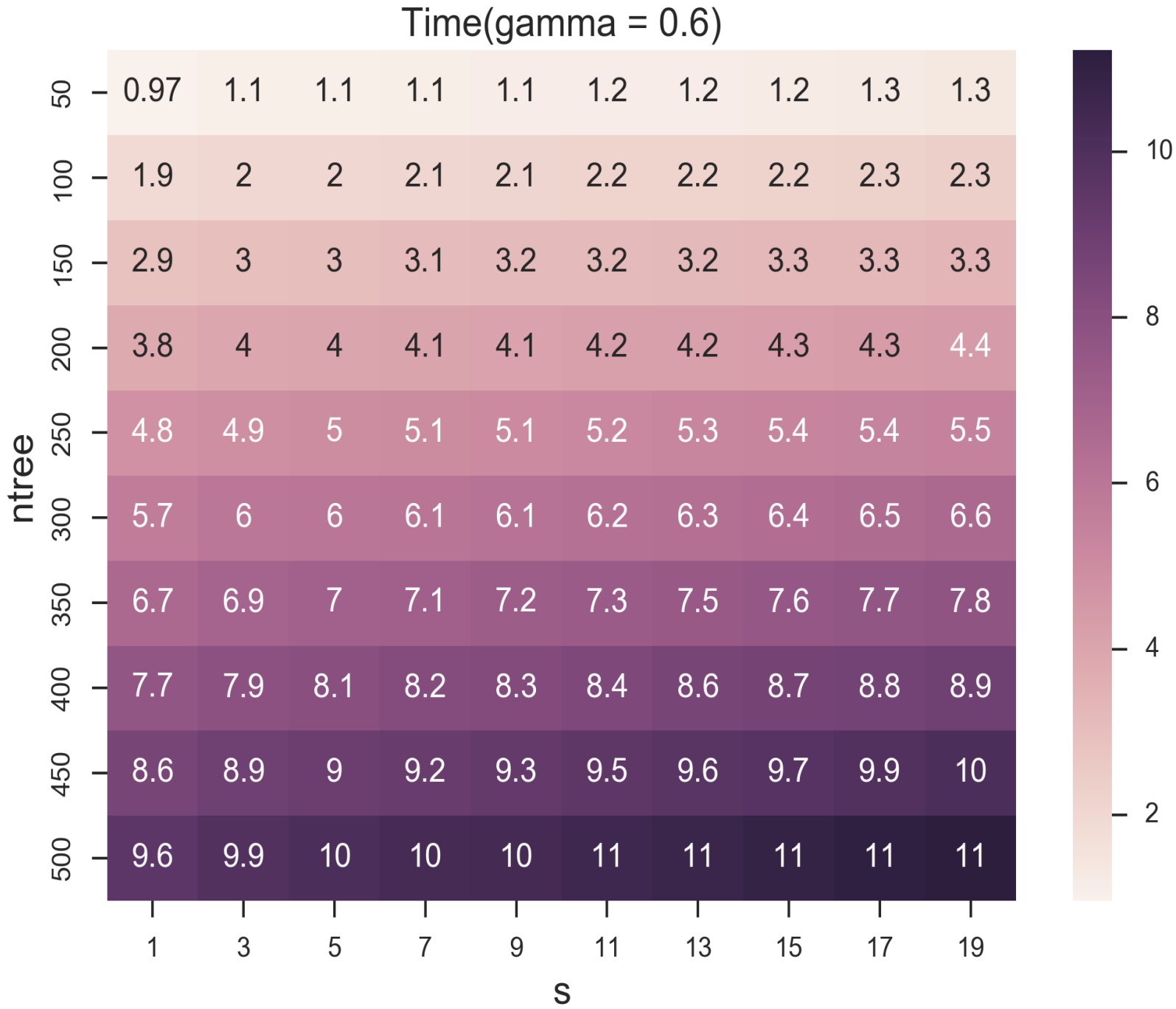


Figure 5. RMSE measures of BLRF algorithm: RMSE ~ s


Time(gamma = 0.6)
ntree
50 0.97 1.1 1.1 1.1 1.1 1.2 1.2 1.2 1.3 1.3
100 1.9 2 2 2.1 2.1 2.2 2.2 2.2 2.3 2.3
150 2.9 3 3 3.1 3.2 3.2 3.2 3.3 3.3 3.3
200 3.8 4 4 4.1 4.1 4.2 4.2 4.3 4.3 4.4
250 4.8 4.9 5 5.1 5.1 5.2 5.3 5.4 5.4 5.5
300 5.7 6 6 6.1 6.1 6.2 6.3 6.4 6.5 6.6
350 6.7 6.9 7 7.1 7.2 7.3 7.5 7.6 7.7 7.8
400 7.7 7.9 8.1 8.2 8.3 8.4 8.6 8.7 8.8 8.9
450 8.6 8.9 9 9.2 9.3 9.5 9.6 9.7 9.9 10
500 9.6 9.9 10 10 10 11 11 11 11 11
1 3 5 7 9 11 13 15 17 19
s
10
8
6
4
2


Performance of BLB-RF vs RF
BLB-RF
RF
Predicted
Actual
0
10
20
30
40
50