Name :- Prateek Mahajan

EE511: Homework 4

Problem 1:-
I have read and understood the general instructions at the top of HW4 and I formally declare that all work I turn in for everything in this course will not contain or involve any cheating at all.

Problem 2:-

a) Please refer to the end of this pdf for the RF, ridge & least square regressions.

~~While RF should typically perform better than ridge & least square regressions as the presence of various trees reduces both its bias & variance, due to a lack of parameter tuning, I suspect my ~~rmse~~ RMSE results are higher than that of ridge & least squares regressions~~

b) ~~Please refer~~

Since the provided dataset is high dimensional, has outliers & appears to be non-linear in nature, I would expect that the random forest model would perform better, as its various trees & structure make it more robust to outliers & overfitting, while also allowing it to represent non-linear relation.

-slips better.
However, due to a lack of parameter-tuning that is not the case here & so, the RMSE of RF is higher than ridge & least squares.

b) Please refer to the end of this pdf for this question.

Problem 3:—

a) The computational complexity of optimizing a tree of depth d for dataset of size n with m features is difficult to generalise & depends on the method of optimisation. Therefore, assuming a binary tree, find some cases below:

- Worst case complexity w/o optimisations: $O(\underbrace{n\log n}_{\text{sorting complexity}} \cdot \underbrace{(mn)}_{\text{traversing through split points}} \cdot \underbrace{2^d}_{\text{checking all case w/o pruning}})$

$$= O(mn^2 \log n \cdot 2^d)$$

- Random Forest: $O(T \cdot n \cdot K \cdot \log n)$, where T = no. of trees
  $K = \sqrt{m}$ for classification & $K = m/3$ for regression
- GBDT: $O(Tmnd)$, where T, m, n & d have meanings as mentioned above

b) The most expensive part of the GBDT computation is finding the split point at a given node. At this stage, we typically need to parse the split nodes of all features to find the ideal one

1 possible optimisation in large datasets is using sparse matrices (particularly for binary classification problems). We could also do pruning of some nodes/trees that ~~wa~~ contribute little to the entire model, thereby reducing computation.
Another possibility is to use histogram-based methods to find the split points.

c) The "split-point" finding part of the computation ~~for the "n" feat~~ of a given node could be parallelised as ~~there~~ are no dependencies ~~between the~~ in this calculation b/w various features.

→ The simplification of the left & right child of a node could be run on separate threads, as again these would be independent of one another / would have no dependencies on each other.

d) There are 2 major differences in how GBDT updates its model parameters vs. GD, SGD & CD:-

→ Each new tree/model in GBDT is given a negative impact of a loss function with respect to the predictions of previous trees in the GBDT. This is unlike the other model, which don't use info from an "ensemble".

→ SGD, GD & CD retain the same number of parameters throughout training. Depending on the data, GBDT may add more parameters during training, as its leaf nodes increase

e) Once again GBDT should perform better than least squares regression & ridge regression. However in this case, the RMSE I got was higher (I'm suspecting due to a lack of hyperparameter tuning). Please refer to the end of the pdf for the data.

f) Please refer to the end of the pdf for this qn.

g) GBDT focusses on reducing ~~var~~ bias by focussing on correcting the output of the previous trees. Random forests, on the other hand, focusses on ~~red~~ reducing variance by using different subsets of data & features.

In the 3 experiments, GBDT & RF gave similar results for both classification problems, but RF outperformed GBDT in the regression problem. I would attribute this to the datasets. Since, the boston dataset, is noisy, reducing variance is more important & so, RF performs better. In the credit-g & breast cancer datasets, GBDT & RF had similar performance, ~~but~~ GBDT edged out RF as the dataset's were complex & reducing bias to prevent overfitting was more important than variance.

Problem 4: —

Please refer to the end of this pdf