CSCI-SHU 360 Homework4

Ninghao Lu nl2752

2024 Fall

Exercise 1 Random Forests

1.1 Solution:

After my implementation, for Boston house price dataset using Random Forests, the training RMSE is 3.39917930348728, the test RMSE is 4.191538495780835.
If we use least square, the training RMSE is 4.822434482543473 and the test RMSE is 5.209217510530889.
If we use ridge regression, the training RMSE is 4.820626531838222 and the test RMSE is 5.186569984437072.
The performance of RF is better on training and test RMSE on Boston ghouse price dataset compared to least square regression and ridge regresson.

1.2 Solution:

Credit-g dataset

For Credit-g dataset using Random Forests, the training accuracy is 0.78, the test accuracy is 0.72666667.

Breast cancer diagnostic dataset

For the cancer diagnostic dataset using Random Forests, the training accuracy is 0.9849246231155779, the test accuracy is 0.9707602339181286.

Exercise 2 Gradient Boosting Decision Trees

2.6.1 Solution:

For every node in the tree, we want to find a pair (p_j, τ_j) so that we can find a feature and threshold for each split.
In front of every node are m (number of features) and n_j (number of samples). For every feature, we first sort and then iterate through every data point to find the best gain, whose time complexity are O($n_j log(n_j)$) and O(n_j) respectively. Therefore, the time complexity for every feature is O($n_j log n_j + n_j$) = O($n_j log n_j$). Since there are m features, the total time complexity for every node is O($m * n_j log n_j$). But if we ignore the sorting step for convenience here, the time complexity is O(m * n_j)
However, at depth d, the data points are split among 2^d nodes. Thus, each node handles $n_j = n/2^d$ nodes. Thus, for depth d, the time complexity is

$$C_d = O(2^d * m * n_j) = O(2^d * m * n/2^d) = O(m * n)$$

Summing up all levels (d in total), the time complexity of optimizing a tree of depth d in terms of m and n is

$$O(m * n * d)$$

2.6.2 Solution:

In GBDT training, the most expensive computation lies in finding the best split at each tree node because we need to consider all the features and all the samples. Therefore, when searching for the best split, instead of considering all the features, we can randomly select a subset of features m' < m for evaluation. This can reduce the number of features that need to be evaluated at each node and increases randomness, which helps prevent overfitting.

2.6.3 Solution:

In GBDT training, the parts where we iterate through each feature and each data point to find the best split can be computed in parallel. In the implementation, I parallelize the process of finding the best feature for each node.

2.6.4 Solution:

For the Boston house price dataset using GBDT, the training RMSE is 1.7378787449656554, the test RMSE is 3.4593227483389075.
If we use least square, the training RMSE is 4.822434482543473 and the test RMSE is 5.209217510530889.
If we use ridge regression, the training RMSE is 4.820626531838222 and the test RMSE is 5.186569984437072.
The performance of GBDT is better on training and test RMSE on Boston ghouse price dataset compared to least square regression and ridge regresson.

2.6.5 Solution:

For the Credit-g dataset using GBDT, the training accuracy is 0.7728571428571429 and the test accuracy is 0.7133333333333334.
For the breast cancer diagnostic dataset, the training accuracy is 0.964824120603015 and the test accuracy is 0.9239766081871345.

2.6.6 Solution:

From the problems above based on my implementation, I found that on the Boston houseprice dataset, GBDT outperforms Random Forests on training and testing RMSE. This is probably because the dataset is continuous-valued, the relationship between features are complex and the noise within the dataset is moderate. Therefore, by correcting the error on made on the previous tree, GBDT can capture the non-linear relationship in the data better and focus more on the hard-to-fit region of the data compared to Random Forest.
However, on the credit-g dataset and breast cancer diagnostic dataset, GBDT performs worse than Random Forests on both training and testing dataset. This is probably because the credit-g dataset and breast cancer dataset may contain noisy features and imbalanced data, and some features may be correlated. Since GBDT focuses more on minimizing residuals, it may try to capture all the noise and even unnecessary features and result in overfitting while by averaging the predictions from multiple independent trees, RF is more robust to these noise and overfitting.