人工智慧導論 HW3 Report

許博翔

May 22, 2024

1 Performance

The following performances are those with the best parameters.

Logistic Regression Accuracy: 0.9333333333333333

Decision Tree Classifier Accuracy: 0.9555555555555556

Random Forest Classifier Accuracy: 0.9777777777777777

Linear Regression MSE: 22.06285648717403

Decision Tree Regressor MSE: 17.4342741702977

Random Forest Regressor MSE: 10.428790772456667

One can see that the performance of using random forest is the best in both classification and regression tasks. The reason is that comparing to the other 2 methods, random forest can deal with outliers better. Outliers don't affect that much to the result of running random forest. The reason that decision tree performs better than the linear model may be that the data is not linearly separable.

2 Normalization and Standardization

Performance of logistic regression (number of iterations=1000, learning rate= 0.01):

Accuracy using normalization: 0.6444444444444445

Accuracy using standardization: 0.8

One can see that standardization performs better than normalization. The reason is that if a feature has an outlier (for example a feature that most data are between 1 and 3 but an outlier is 100), then normalization will significantly reduce the impact of this feature (comparing to other features), but standardization won't.

3 Learning Rate and Number of Iterations

3.1 Logistic Regression

Learning rate=0.01, number of iterations=1000: 0.8

Learning rate=0.01, number of iterations=10000: 0.9333333333333333

Learning rate=0.1, number of iterations=1000: 0.9333333333333333

One can see that both increasing the number of iterations and increasing the learning rate will increase the performance. The reason is that small learning rate or few number of iterations is not enough to fit the model.

3.2 Linear Regression

The result of directly using the pseudo inverse to calculate it: 22.33701156362598

Learning rate=0.01, number of iterations=1000: 22.06285648717403

Learning rate=0.01, number of iterations=10000: 22.337009206623446

Learning rate=0.1, number of iterations=1000: 22.337009379619968

One can see that both increasing the number of iterations and increasing the learning rate will make the resulting $\mathbf{w}$ closer to the optimal solution (which can be get by using the pseudo inverse). However, the closer to the optimal solution does not imply the smaller the MSE is because it may overfit the training data.

4 The Number of Trees and Maximum Depth

4.1 Classification

The number of trees=100, maximum depth=5: 0.9777777777777777

The number of trees=500, maximum depth=5: 0.9777777777777777

The number of trees=100, maximum depth=10: 0.9777777777777777

The number of trees=10, maximum depth=5: 0.9777777777777777

The number of trees=100, maximum depth=2: 0.9777777777777777

One can see that the number of trees= 10 with maximum depth 5 or the number

of trees= 100 with maximum depth 2 is enough to fit the training data.

4.2 Regression

The number of trees=100, maximum depth=5: 10.70623317539579

The number of trees=500, maximum depth=5: 10.86428686562275

The number of trees=100, maximum depth=10: 10.428790772456667

One can see that the MSE increases when we increase the number of trees. The reason is that although more trees implies better performance on the training data, it may also overfit and thus performs worse on the testing data.

One can also see that the MSE decreases when we increase the depth of trees. The reason is that trees with maximum depth=5 are not enough to fit the training data.

5 Strengths and Weaknesses of Models

Strengths:

Linear: easy, straightforward, good performance when data is linearly separable.

Nonlinear: easier and faster than random forest, perform better when data is not linearly separable

Random forest: perform really well with some outliers, often perform better than the other two models

Weaknesses:

Linear: perform bad when data is not linearly separable

Nonlinear: perform worse than random forest if there are some outlier, easy to overfit if the tree is too deep

Random forest: time consuming, easy to overfit

If the data is approximately linearly separable, then I'll choose linear model. If I have few computing resources, I'll choose nonlinear model. Otherwise, in mose case, I'll choose random forest. Since in real world, there are usually some outliers in data, and random forest can deal with that well.