Reflection for Assessment2

Yewei Y, Dec 2019

Firstly, our team discussed together and decided how to complete this project. We found that the classification algorithms are better than the clustering algorithms in the last assessment in our work, and the tree-based classification algorithms has the best effect. So, in this assessment our team focused on tree algorithms. And in last time I have tried doing some multiple classification, so I think this time we can extended the problem of binary classification and considered the network attack types such as dos. The baseline is clearly a decision tree as we would all use tree-based classification algorithms, we want to see an improvement.

Different from the last time, I worked closely with my team members on this project. In this project, I was mainly responsible for preprocessing the data and providing the team members with the processed training and test sets. Then provide the XGBoost model for deeply research. Tianhang and Linlan responsible separately baseline and further research except another two tree-based models. The work of each member of our group is needed for each other.

Through communication with team members, both two parts I responsible for have improved compared to the last time ：

1.Data preprocessing:

Last time I did a lot work to pre-processing like variance analysis, correlation analysis, outlier analysis, normalization and dummy variables. They are effective but not efficient.

Considering in last time I used a random forest to rank the importance in my project, Linglan suggested that I can use this directly for preprocessing so that I do not need to use variance analysis and correlation analysis. Then I got an idea that repeating times of random forest on original dataset and pick out the primary component. I looked for important features through the random forest five times and select the first few vectors with the sum of importance ratio of 95% (I want to remain most of information). After experiments, 19 vectors were selected. It is effective and efficient, and the result is beautiful and similar to the last project I did.

Moreover, normalization is not needed because our models are tree-based. And I also found dummy variable are not suitable for this time. The main reason is it will add a lot of dimensions and make variable information. Actually, I have tried using the dummy variables in the beginning random forest. Because of the processing upper limit size, I have to delete some unimportant factors. The result is terrible. Finally, I decided to directly number the factors.

To meet research needs, I used kddnames.txt to classify various attacks into 5 categories: "normal", "dos", "probe", "u2r", "r2l". At last, I randomly separated 6 classes (5 abnormal and 1 normal) dataset with 70% of the training set and 30% of the test set then gave them to my teammates.

2.Model:

This time I used XGBoost. It is currently the fastest and best open source boosted tree toolkit. The algorithm applied by XGBoost is an improvement of GBDT (gradient boosting decision tree), Linlan's work, which can be used for both classification and regression problems. Linlan and I research the difference in the further research in the report.

The objective function (loss function and regularization) of XGBoost at iteration t that we need to minimize is the following [1]:

Real value (label) known from the training data-set

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(\mathbf{x}_i)) + \Omega(f_t)$$

Can be seen as f(x + Δx) where x = $\hat{y}_i^{(t-1)}$

My understanding is that objective function is loss function (fit as best as possible) plus regular term (less complexity), loss function is the difference between the predicted value and the real value.

It is easy to see that the XGBoost objective is a function of functions (i.e. l is a function of CART learners).

Then we need Taylor approximation to transform the original objective function to a function in the Euclidean domain, in order to be able to use traditional optimization techniques. Traditional GBDT only uses first-order derivative for optimization, and XGBoost performs second-order Taylor expansion of the loss function and uses first- and second-order derivatives at the same time. The algorithm has been written in the report.

I created a model and looked the result and found my result of importance is very different from Tianhang and Linlan's result. So, I drew some scatter plots to help analysising the features. The most interesting one has given Linlan to analysis in the report (Linlan did the feature analysis work in our team, she found lots of features by kinds of pictures), which compare the most important feature of mine and theirs. By the way, it's worth celebrating XGBoost worked best after our analysis (by the comparison of confusion matrix and time complexity), thanks to it is an improvement of GBDT.

Finally, thanks for all the help from my team members.

[1] XGBoost Mathematics Explained, https://towardsdatascience.com/xgboost-mathematics-explained-58262530904a