

# Bayesian Optimization Report

Hongfei Li

I use Bayesian Optimization to tune the hyperparameters of random forest. In STAT 542 of Spring 2017 my team did a project of predicting default using the lending club data. At that time, we used a random forest model with two hyperparameters: the number of trees = 400, and the number of features to consider at each split = 7. We had a test log loss of 0.08628.

The original project was done in R. I took the preprocessed data and wrote a Python script for the tuning process. The preprocessed data has 35 columns. 687379 rows of train data. And 200000 rows of test data.

The two hyperparameters are:

max_feature	the number of features to consider at each split, represented as a ratio between (0,1)
n_estimator	the number of trees

I use Expected Improvement as the acquisition function. The target that we would like to minimize is the 5-fold cross-validation log loss, that is equivalent to maximize the negative log loss.

The cross-validation log loss of the original model is 0.3294.

In my first attempt, I choose the interval as:

max_feature	[0.1, 1)
n_estimator	[200, 800]

Initialization				
Step	Time	Value	max_features	n_estimators
1	170m27s	-0.410226	0.7310	751.5816
2	108m35s	-0.428598	0.6906	499.3653
3	76m28s	-0.365762	0.4235	618.9109

Bayesian Optimization				
Step	Time	Value	max_features	n_estimators
4	06m00s	-0.240168	0.1000	200.0000
5	79m42s	-0.577928	0.9990	256.1019
6	24m11s	-0.223014	0.1000	800.0000
7	11m52s	-0.227418	0.1000	398.8062
8	118m11s	-0.523216	0.9990	368.3534
9	13m19s	-0.229964	0.1000	426.4985
10	13m05s	-0.228084	0.1000	410.9360
11	21m04s	-0.223856	0.1000	682.2315

12	16m58s	-0.224016	0.1000	562.5149
----	--------	-----------	--------	----------

Then I shrink the interval:

max_feature	[0.1, 0.3]
n_estimator	[log(200), log(800)]

Initialization				
Step	Time	Value	max_features	n_estimators
1	21m18s	-0.224576	0.1043	6.5786
2	17m31s	-0.292748	0.1522	5.9468

Bayesian Optimization				
Step	Time	Value	max_features	n_estimators
3	70m29s	-0.326704	0.3000	6.6846
4	05m39s	-0.246032	0.1000	5.2983
5	17m09s	-0.392904	0.3000	5.2983
6	15m48s	-0.225162	0.1000	6.3269
7	22m29s	-0.223254	0.1000	6.6846

The best set of hyperparameters is max\_features=0.1, n\_estimators=800. The test log loss of this selected model is 0.07988, which is better than the original model (0.08628).

It seems that, less features and more trees are better.