



Completed • Knowledge • 1,694 teams

Forest Cover Type Prediction

Fri 16 May 2014 – Mon 11 May 2015 (4 months ago)

Dashboard ▾

Competition Forum

[All Forums » Forest Cover Type Prediction](#)

Search

◀ Prev
Topic

Caret - GBM - Cross Validation - Takes an extremely long time to complete

Next
Topic ▶

Start Watching

▲
0
▼

Hi Guys, My name is Abhi. I am trying to use gbm in the caret package for this problem. I want to do some cross validation to get the ideal values for the tuning parameters but the scripts are taking an extremely long time to complete (most of the time I have to kill R). Here is my code.

```
myTuneGrid <- expand.grid(n.trees = 1:500, interaction.depth = 1:22, shrinkage = 0.1)
```

```
fitControl <- trainControl(method = "repeatedcv", number = 7, repeats = 1, verboseIter = FALSE, returnResamp = "all")
```

```
myModel <- train(Cover_Type ~ ., data = modelData, method = "gbm", trControl = fitControl, tuneGrid = myTuneGrid)
```

Am I doing something wrong here? Any tricks to optimize the performance? Any help would be appreciated

#1 | Posted 11 months ago

[abhimanipal](#)▲
0
▼

Abhi,

I would start with the interaction depth - it's likely that an interaction depth of 22 doesn't make much sense. Start with something like 2:5, and check the results. If the model performance increases with higher interaction depths, expand to say 5:10 etc etc.

You could also speed up things by using 3-fold CV instead of 7 - each fold would be trained on less data (2/3 of the training data, instead of 6/7). Once you've tuned the parameters, you could always go back to 7-fold CV for a more accurate estimate, but it's possible that 3-fold CV is already accurate enough.

To summarize : start smaller

#2 | Posted 11 months ago

[Rudi Kruger](#)

▲

Thank you. That is very helpful

0



#3 | Posted 11 months ago



abhimanipal



0



Coming across this post rather late but I noticed something else that might be useful to you or others who are experiencing similar problems.

The way you have your grid space set up in the original post results in 11,000 combinations of parameters, largely because you are trying every integer value of n.trees from 1-500. I feel like you could perform a lower resolution scan of that range and greatly reduce the number of models to fit. For example, replacing n.trees=1:500 with n.trees=seq(1,501,10) would reduce the number of parameter combinations by a factor of 10. Then if you like, you can go back and re-run with higher resolution for n.trees over a range that produces the best results.

#4 | Posted 11 months ago



schwartz



0



Thank you that's an excellent suggestion. Much appreciated

#5 | Posted 11 months ago



abhimanipal

Reply

B *I*[+ Add attachment\(s\) ...](#)**Post Reply**☒ Email me when someone replies[Start Watching](#)[« Back to forum](#)