

Lab #6: Boosting Algorithms

Biostat 216: Machine Learning in R for the Biomedical Sciences

Classification Trees

1. Install the `gbm` package
2. Load the `gbm` package
3. Read in the dementia data “dementia2.csv” into a data frame called `dat`
4. Set a random seed of 3 and select 330 rows at random to be the training set and call the set of indices `train`. Select 165 of the remaining subjects to be the `validation` set, and the final 165 will be the `test` set.
5. Try to fit a logistic generalized boosted regression model with `Dementia` as response and all other variables as predictors except for `MacCohort_kr` on the training data. Run the model for 10000 trees, a shrinkage parameter (`lambda`) of 0.01, and tree (interaction) depth of 3. Check the help for `gbm` to determine whether the default parameters are appropriate and if not change the ones that you need.
6. Convert the `Dementia` variable to be 0/1 (0=“No”, 1=“Yes”) – and also regenerate `datTrain`, `datValidation` and `datTest`.
7. Retry the logistic generalized boosted regression model with `Dementia` as response and all other variables as predictors except for `MacCohort_kr`. Then run the `summary` command on the fitted model.
8. Run `plot(gbDemLogistic,i="Left_MTG_middle_temporal_gyrus",type="response")`
9. Now try a similar plot for the weaker predictor of `gm`
10. And the same again for the even weaker `Right_TTG_transverse_temporal_gyrus`
11. Rerun `gbm` with the “adaboost” distribution instead of “bernoulli”. Then run the `summary` command on the fitted model.
12. Do you get the same most important predictor as in the logistic version?
13. What about the same 2nd predictor?
14. Make a prediction on the validation set. In boosting, the number of trees is a tuning parameter, and if we use too many trees it is possible to overfit. The number of trees should generally be selected with cross-validation. For the lab we will do the simpler, but less effective, approach of using the validation error to determine the best number of trees and will plot the relationship between number of trees and validation error. Make a sequence `nTrees` from 100 to 10000 in steps of 100.
15. Examine the fit of the Adaboost model on the validation data with the `predict` command and the argument `n.trees=nTrees`
16. Check the dimensions of the resulting object
17. Estimate and plot the classification error rate in the validation data against the number of trees.
18. Find the best number of trees based on those examined and determine its classification error in the validation set.
19. Now find the classification error for this number of trees in the test set.
20. Comment on the difference in error estimates between the validation and test data for the best number of trees found in the validation error.