Evaluate testing data (multi-class) - Lasso

Andrew Chang

2018-12-05

Contents

Note: The two differences between Lasso and Tree-based methods are:

1. Lasso does not output 'size', so the "number of feature used"" to train the model is not informative. Should check 'lambda' which is correspond to the number of features used.

2. Lasso's vimp will be based on how many times the feature exist in all runs.

Labels: 0: Basal 1: LumA

```
## user input
project_home <- "~/EVE/tests"
project_name <- "lasso_binary_outCV_test"
```

0. Load Data

```
## 199 of samples were used

## 100 of full features

## 2 runs, each run contains 3 CVs.

## Labels:

##
##   0   1
##  50 149
```

1. Scores

`average = T`: scores are based on pre-validation (combine predictions from all CVs per job and then calculate a single score per job).

`average = F`: report all the scores from each CV across entire jobs.

Note: '# of feature retain' is not meaningful in this setup. It is just the total number of features. The RFE concept is cooked into 'lambda' optimization, and each CV may have different lambda values, so it makes comparing prevalidation scores at different lambda difficult.
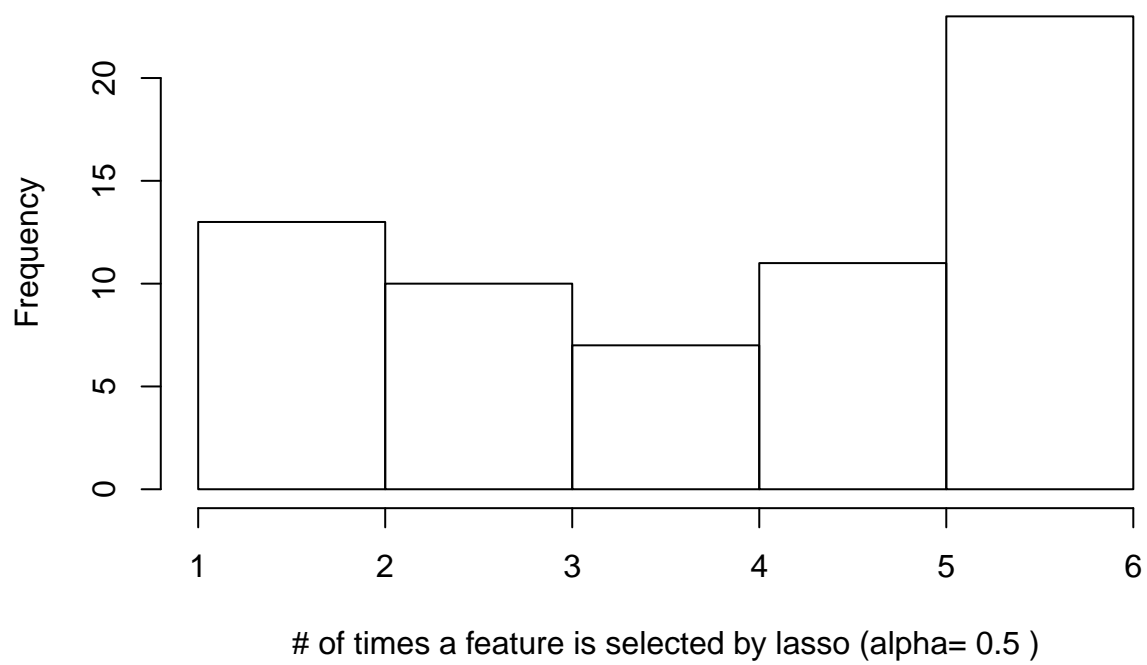
Averaged scores during RFE

Averaged scores during RFE
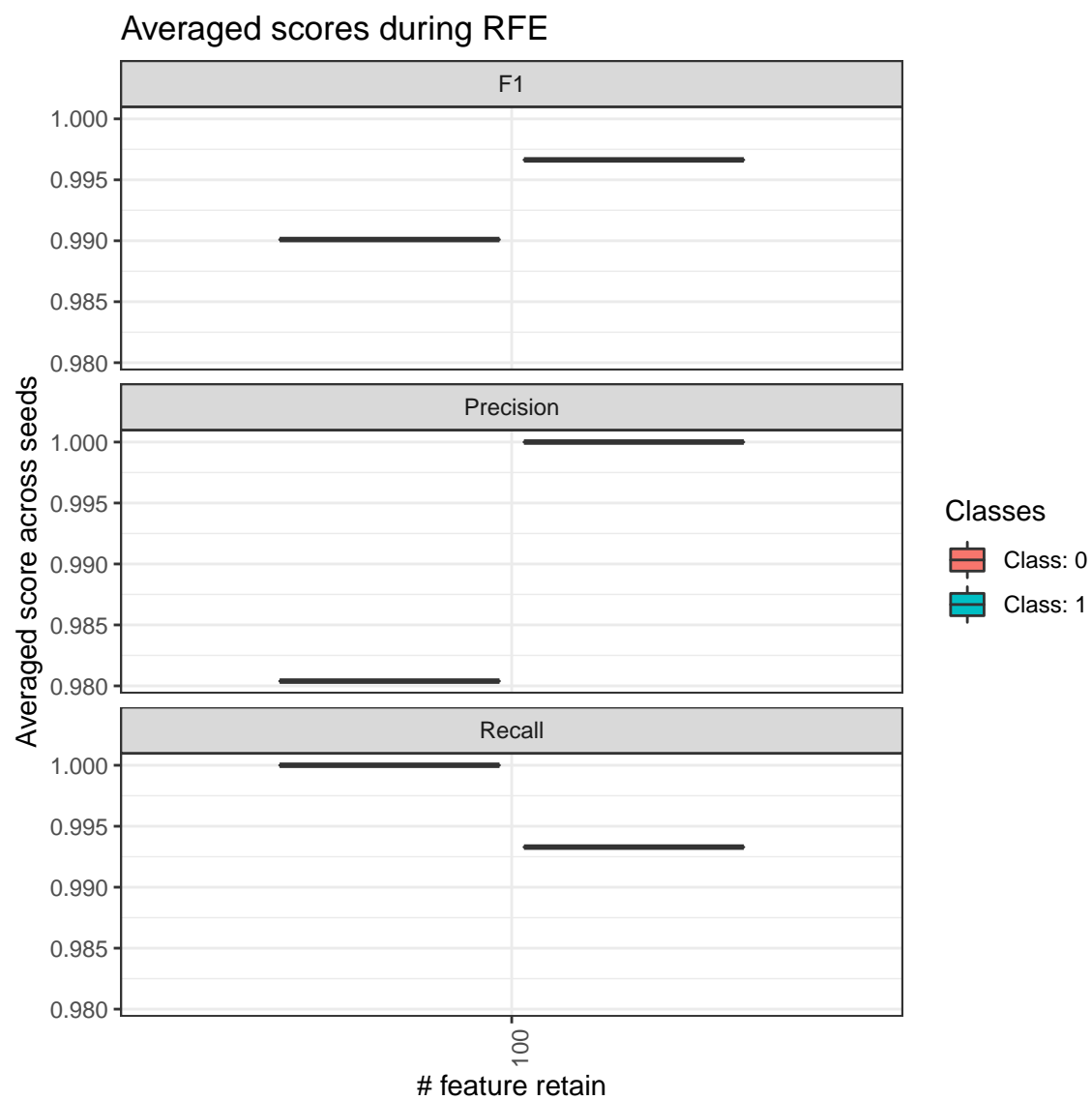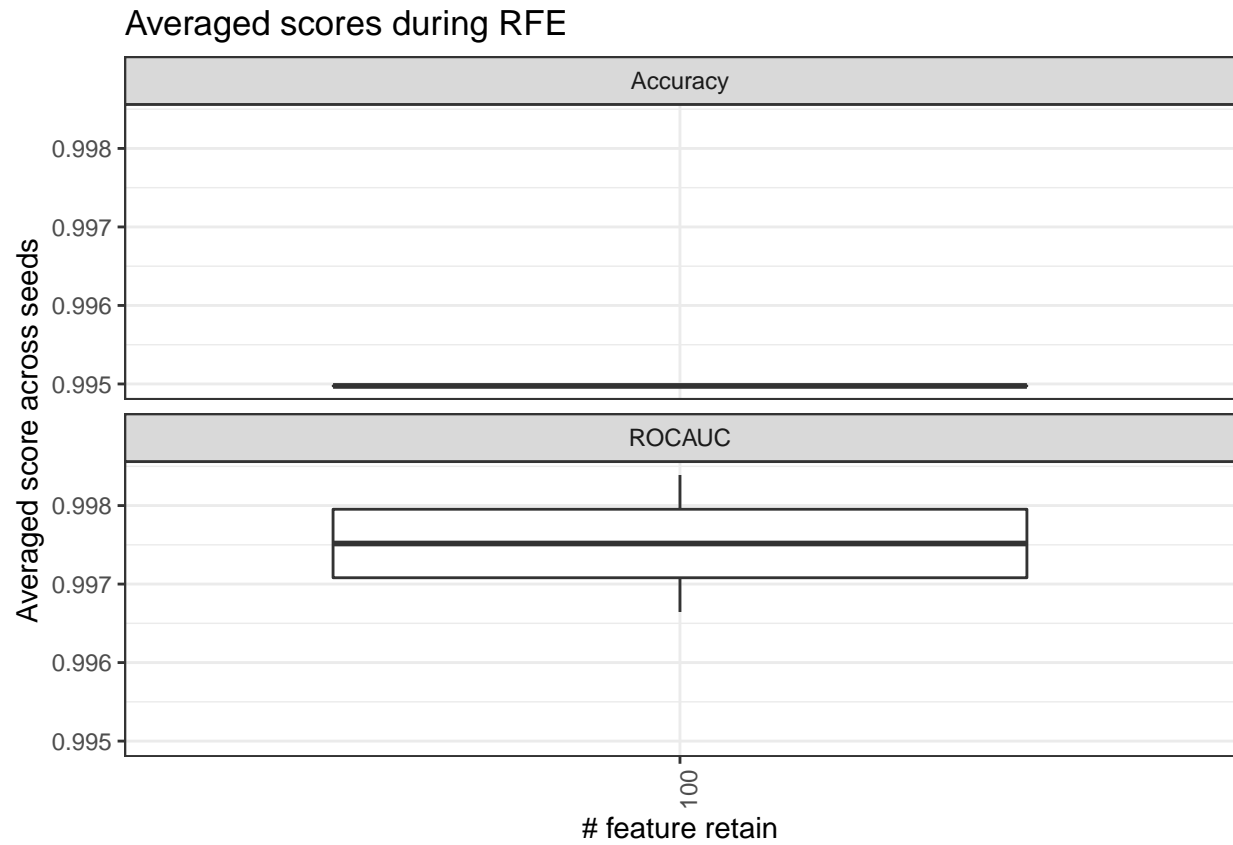
2. Important Features

For Lasso, we calculate how many times a given features is being used in all the runs.

distribution across 2 seed x 3 CV

# of times a feature is selected by lasso (alpha= 0.5)

most used features

GeneID.84281
GeneID.8200
GeneID.81706
GeneID.80129
GeneID.79083
GeneID.7832
GeneID.771
GeneID.7494
GeneID.7272
GeneID.55839
GeneID.55259
GeneID.53335
GeneID.4137
GeneID.319089
GeneID.27202
GeneID.2625
GeneID.23397
GeneID.2296
GeneID.221061
GeneID.2099

0 1 2 3 4 5 6