Lasso CV

Tucker Morgan - tlm2152

3/20/2022

```
library(tidyverse)
```

```
## Warning: package 'dplyr' was built under R version 4.1.3
```

```
library(glmnet)
```

```
bc <-
  read_csv("./data/breast-cancer.csv") %>%
  mutate(diagnosis = 1 *(diagnosis == "M")) %>%
  select(-id)

source("./shared_code/partition.R")

part_bc <- partition(p = 0.8, data = bc)

bc_trn <-
  part_bc %>%
  filter(part_id == "train") %>%
  select(-part_id)

bc_tst <-
  part_bc %>%
  filter(part_id == "test") %>%
  select(-part_id)
```

```
source("./shared_code/cv_folding.R")

bc_trn_folds <-
  cv_sets(training = bc_trn) %>%
  select(-fold_p)
```

```
set.seed(100)

X <- bc_trn_folds[, -c(1,32)]
X <- as.matrix(X)
Y <- bc_trn_folds$diagnosis
lambda_vec <- seq(0, 0.4, length = 5) # lambda vector for testing

# creating a simple example function for testing
ex_func <- function(x, y, lambda_vec){
```

```
  glmnet(x = x, y = y,
         standardize = TRUE,
         alpha = 1,
         lambda = lambda_vec,
         family = "binomial"(link = "logit"))
}

ex_func(x = X, y = Y, lambda_vec = 0) %>% coef() # just for example, not stored
```

```
## 31 x 1 sparse Matrix of class "dgCMatrix"
##                                    s0
## (Intercept)             -5.377793e+02
## radius_mean             -4.606398e+01
## texture_mean             3.422739e+00
## perimeter_mean           1.086056e+01
## area_mean               -2.455698e-01
## smoothness_mean         -5.544642e+01
## compactness_mean        -1.157854e+03
## concavity_mean           9.011997e+02
## concave points_mean      1.240681e+03
## symmetry_mean           -8.837807e+02
## fractal_dimension_mean   4.847768e+03
## radius_se                5.422879e+02
## texture_se              -2.330474e+01
## perimeter_se            -9.614294e+00
## area_se                 -1.941708e+00
## smoothness_se            6.665369e+03
## compactness_se          -8.276168e+02
## concavity_se            -1.246332e+03
## concave points_se        8.536210e+03
## symmetry_se             -1.020098e+04
## fractal_dimension_se    -1.070869e+04
## radius_worst            -1.507424e+01
## texture_worst            3.458008e+00
## perimeter_worst         -2.169199e+00
## area_worst               3.995993e-01
## smoothness_worst        -7.442752e+02
## compactness_worst       -1.413013e+02
## concavity_worst          2.031756e+02
## concave points_worst     3.060762e+01
## symmetry_worst           1.382811e+03
## fractal_dimension_worst -9.031451e+02
```

```
cv_function <- function(k = 5, training, func, lambda_vec){

  auc_list = list()
  mean_auc_list = list()
# first, a for loop to iterate over a lambda vector
  for (j in 1:length(lambda_vec)){
    # and now we have a for loop to iterate over each fold, k = 5 here
    for (i in 1:k){
      # this will identify the training set as not i
      trn_set =
```

```
      training %>%
      filter(fold_id != i) %>%
      select(-fold_id)
    # and this assigns i to be the test set
    tst_set =
      training %>%
      filter(fold_id == i) %>%
      select(-fold_id)
    # making matrices
    X_trn <- as.matrix(trn_set[,-1])
    X_tst <- as.matrix(tst_set[,-1])
    Y_trn <- trn_set$diagnosis
    # fitting our function based on training set
    trn_fit = func(x = X_trn, y = Y_trn, lambda_vec = lambda_vec[j])
    # calculating AUC
    trn_pred <- predict(trn_fit,
                        newx = X_tst,
                        type = "response")
    trn_roc <- pROC::roc(tst_set$diagnosis, trn_pred)

    auc_list[[i]] = trn_roc$auc
  }
  # calculating mean cv auc for each lambda
  auc_df = data.frame("auc" = do.call(rbind, auc_list))
  mean_auc = mean(auc_df$auc)
  mean_auc_list[[j]] = data.frame("mean_auc" = mean_auc, "lambda" = lambda_vec[j])
}
# creating dataframe to show lambda values and corresponding mean AUC
res = as.data.frame(do.call(rbind, mean_auc_list))

return(res)
}

cv_function(training = bc_trn_folds, func = ex_func, lambda_vec = lambda_vec)
```

```
##    mean_auc lambda
## 1 0.9815993    0.0
## 2 0.9883679    0.1
## 3 0.9851657    0.2
## 4 0.9848944    0.3
## 5 0.5000000    0.4
```

The cross-validation function seems to work as intended, but I found that the mean AUC dropped to 0.5 (no discrimination) at a seemingly low lambda value of 0.4. I'll run `glmnet` below to show that all coefficients drop out at lambda = 0.4.

```
glmnet(x = X, y = Y,
       standardize = TRUE,
       alpha = 1,
       lambda = 0.4,
       family = "binomial"(link = "logit")) %>%
  coef()
```

```
## 31 x 1 sparse Matrix of class "dgCMatrix"
##                                s0
## (Intercept)            -0.5953494
## radius_mean             .
## texture_mean            .
## perimeter_mean          .
## area_mean               .
## smoothness_mean         .
## compactness_mean        .
## concavity_mean          .
## concave points_mean     .
## symmetry_mean           .
## fractal_dimension_mean  .
## radius_se               .
## texture_se              .
## perimeter_se            .
## area_se                 .
## smoothness_se           .
## compactness_se          .
## concavity_se            .
## concave points_se       .
## symmetry_se             .
## fractal_dimension_se    .
## radius_worst            .
## texture_worst           .
## perimeter_worst         .
## area_worst              .
## smoothness_worst        .
## compactness_worst       .
## concavity_worst         .
## concave points_worst    .
## symmetry_worst          .
## fractal_dimension_worst .
```

Bizarre!