Module 2: Supervised Learning

Jianhui Gao, Siyue Yang, and Jesse Gronsbell

31/05/2022

```
# If a package is installed, it will be loaded. If any
## are not, the missing package(s) will be installed
## from CRAN and then loaded.

## First specify the packages of interest
packages <- c(
  "dplyr", "PheCAP", "glmnet", "randomForestSRC", "PheNorm",
  "MAP", "pROC", "mltools", "data.table", "ggplot2", "parallel"
)

## Now load or install&load all
package.check <- lapply(
  packages,
  FUN = function(x) {
    if (!require(x, character.only = TRUE)) {
      install.packages(x, dependencies = TRUE)
      library(x, character.only = TRUE)
    }
  }
)

# load environment from example 1
load("environment.RData")
```

Prepare data for algorithm development

- Split data into training and testing set
- Training 50%, Testing 50%

```
data <- PhecapData(PheCAP::ehr_data, "healthcare_utilization", "label", 0.5,
  patient_id = "patient_id", seed = 123
)

# Transform Features log(x + 1)
labeled_data <- ehr_data %>% dplyr::filter(!is.na(label))


# All Features
all_x <- ehr_data %>% dplyr::select(
  starts_with("COD"), starts_with("NLP"),
  starts_with("main"), healthcare_utilization
)
```

```r
health_count <- ehr_data$healthcare_utilization

# Training Set
train_data <- ehr_data %>% dplyr::filter(patient_id %in% data$training_set)
train_x <- train_data %>%
  dplyr::select(
    starts_with("COD"), starts_with("NLP"),
    starts_with("main"), healthcare_utilization
  ) %>%
  as.matrix()
train_y <- train_data %>%
  dplyr::select(label) %>%
  pull()

# Testing Set
test_data <- ehr_data %>% dplyr::filter(patient_id %in% data$validation_set)
test_x <- test_data %>%
  dplyr::select(
    starts_with("COD"), starts_with("NLP"),
    starts_with("main"), healthcare_utilization
  ) %>%
  as.matrix()
test_y <- test_data %>%
  dplyr::select(label) %>%
  pull()
```

Penalized logistic regression

- Fit LASSO and Adaptive LASSO(ALASSO)

```r
# Choose best lambda using CV
beta.lasso <- lasso_fit(x = train_x, y = train_y,
                        tuning = "cv", family = "binomial")
```

```r
# Features Selected
names(beta.lasso[abs(beta.lasso)>0])[-1]
```

```
## [1] "NLP304"                 "main_NLP"               "main_ICDNLP"
## [4] "healthcare_utilization"
```

```r
# prediction on testing set
y_hat.lasso <- linear_model_predict(beta = beta.lasso, x = test_x,
                                    probability = TRUE)
```

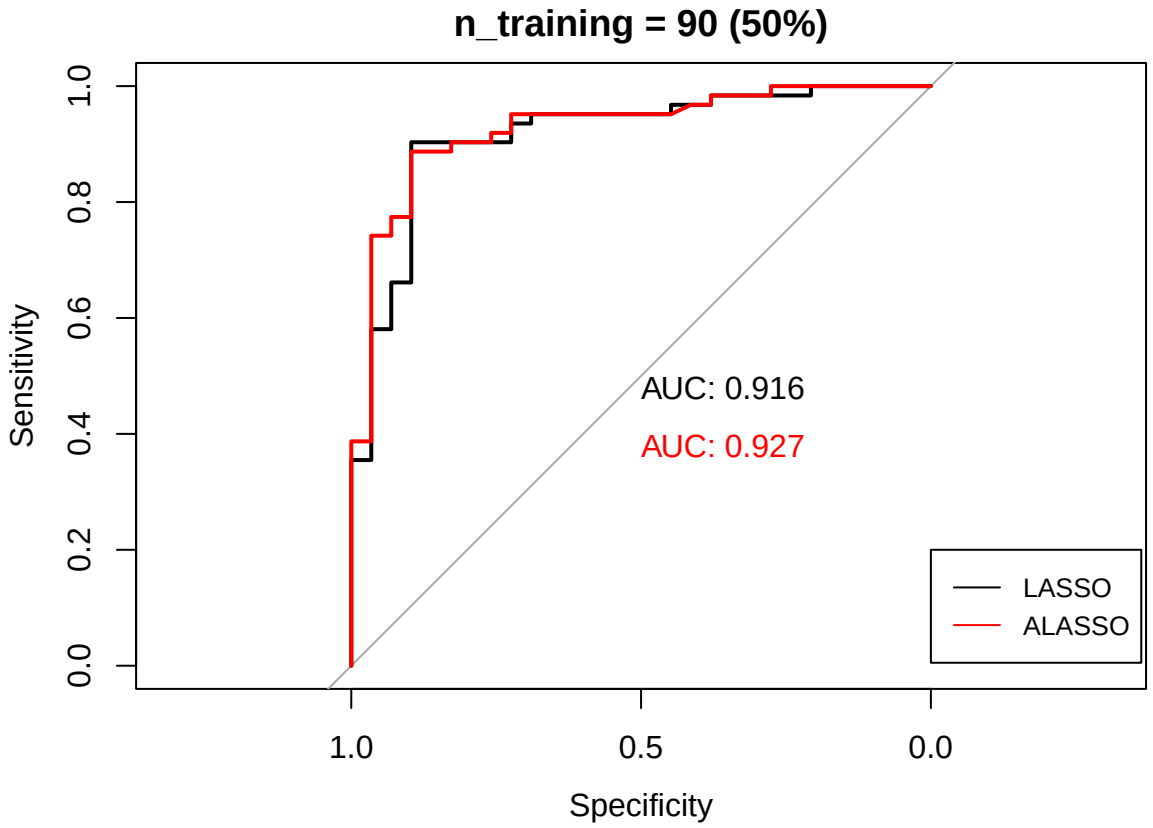
```r
# Fit Adaptive LASSO
beta.alasso <- adaptive_lasso_fit(x = train_x, y = train_y,
                                  tuning = "cv", family = "binomial")
y_hat.alasso <- linear_model_predict(beta = beta.alasso, x = test_x,
                                    probability = TRUE)


# Features Selected
names(beta.alasso[abs(beta.alasso)>0])[-1]
```

```
## [1] "NLP304"                 "main_ICDNLP"            "healthcare_utilization"
```

```
roc.lasso <- roc(test_y, y_hat.lasso)
roc.alasso <- roc(test_y, y_hat.alasso)

plot(roc.lasso,
  print.auc = TRUE, main = "n_training = 90 (50%)"
)
plot(roc.alasso,
  print.auc = TRUE, col = 'red', add = TRUE, print.auc.y = 0.4
)
legend(0, 0.2, legend = c("LASSO", "ALASSO"), col = c("black","red"),
       lty = 1, cex = 0.8)
```




```
roc_full.lasso <- get_roc(y_true = test_y, y_score = y_hat.lasso)
head(roc_full.lasso,10)
```

```
##          cutoff    pos.rate        FPR       TPR       PPV       NPV        F1
##  [1,] 0.9352770 0.005494505 0.00000000 0.1697055 1.0000000 0.3603458 0.2901679
##  [2,] 0.8896985 0.098901099 0.00000000 0.2770827 1.0000000 0.3928428 0.4339308
##  [3,] 0.8562686 0.252747253 0.03448276 0.3729032 0.9585406 0.4186603 0.5369252
##  [4,] 0.8547292 0.252747253 0.03448276 0.4383871 0.9645138 0.4457179 0.6027944
##  [5,] 0.8365560 0.307692308 0.03448276 0.5038710 0.9689826 0.4765146 0.6629881
##  [6,] 0.8048679 0.406593407 0.03448276 0.5693548 0.9724518 0.5118830 0.7182096
##  [7,] 0.7964701 0.417582418 0.06896552 0.6000000 0.9489796 0.5212355 0.7351779
##  [8,] 0.7914085 0.428571429 0.06896552 0.6233871 0.9507995 0.5362463 0.7530443
##  [9,] 0.7732080 0.472527473 0.06896552 0.6467742 0.9524941 0.5521472 0.7704131
## [10,] 0.7583238 0.483516484 0.10344828 0.6879032 0.9342826 0.5733186 0.7923827
```

```
roc_full.alasso <- get_roc(y_true = test_y, y_score = y_hat.alasso)
head(roc_full.lasso,10)
```

```
##          cutoff    pos.rate        FPR       TPR       PPV       NPV        F1
##  [1,] 0.9352770 0.005494505 0.00000000 0.1697055 1.0000000 0.3603458 0.2901679
##  [2,] 0.8896985 0.098901099 0.00000000 0.2770827 1.0000000 0.3928428 0.4339308
##  [3,] 0.8562686 0.252747253 0.03448276 0.3729032 0.9585406 0.4186603 0.5369252
##  [4,] 0.8547292 0.252747253 0.03448276 0.4383871 0.9645138 0.4457179 0.6027944
##  [5,] 0.8365560 0.307692308 0.03448276 0.5038710 0.9689826 0.4765146 0.6629881
##  [6,] 0.8048679 0.406593407 0.03448276 0.5693548 0.9724518 0.5118830 0.7182096
##  [7,] 0.7964701 0.417582418 0.06896552 0.6000000 0.9489796 0.5212355 0.7351779
##  [8,] 0.7914085 0.428571429 0.06896552 0.6233871 0.9507995 0.5362463 0.7530443
##  [9,] 0.7732080 0.472527473 0.06896552 0.6467742 0.9524941 0.5521472 0.7704131
## [10,] 0.7583238 0.483516484 0.10344828 0.6879032 0.9342826 0.5733186 0.7923827
```

Different train size

- randomly sample traning size = 50, 70, 90
- rest as testing set
- repeat 600 times

```
start<- Sys.time()
auc_supervised <- validate_supervised(dat = labeled_data, nsim = 600,
                                      n.train = c(50, 70, 90))
end <- Sys.time()
end - start
```

```
## Time difference of 3.296876 mins
```

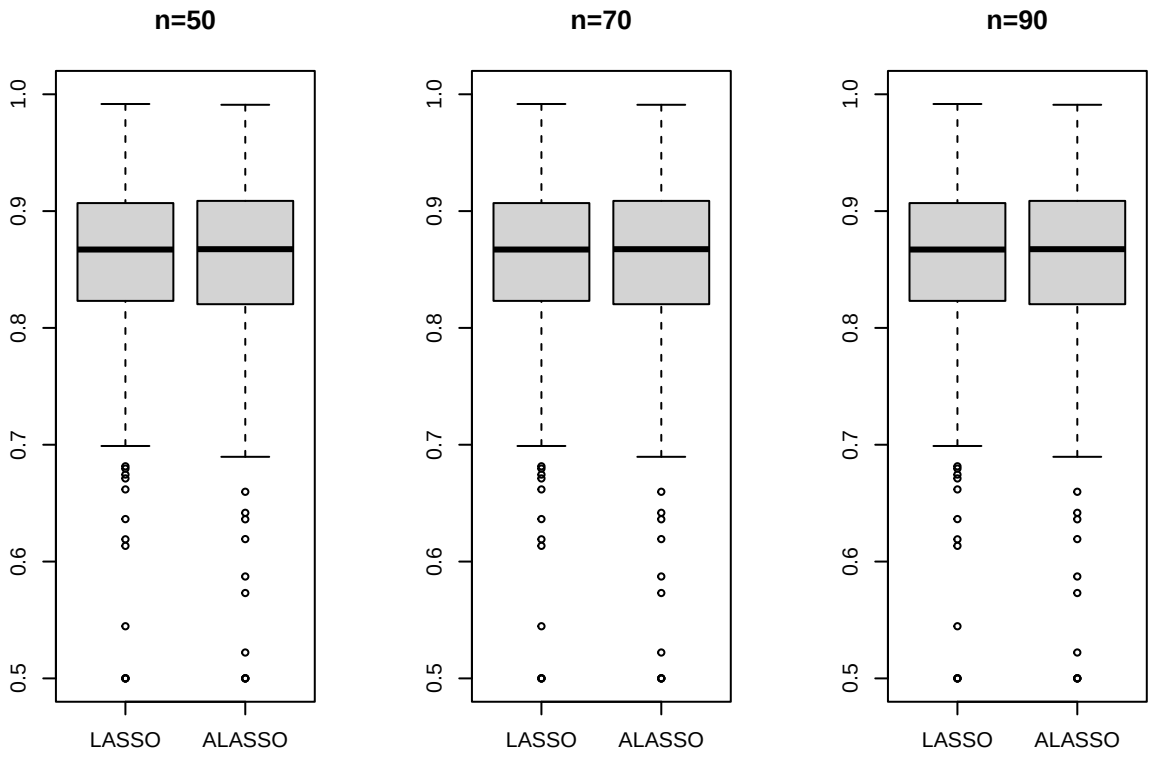
```
# median AUC
apply(auc_supervised, 2, median)
```

```
##  n=50,LASSO  n=70,LASSO  n=90,LASSO n=50,ALASSO n=70,ALASSO n=90,ALASSO
##   0.8670982   0.8789683   0.8907670   0.8673935   0.8736602   0.8855655
```

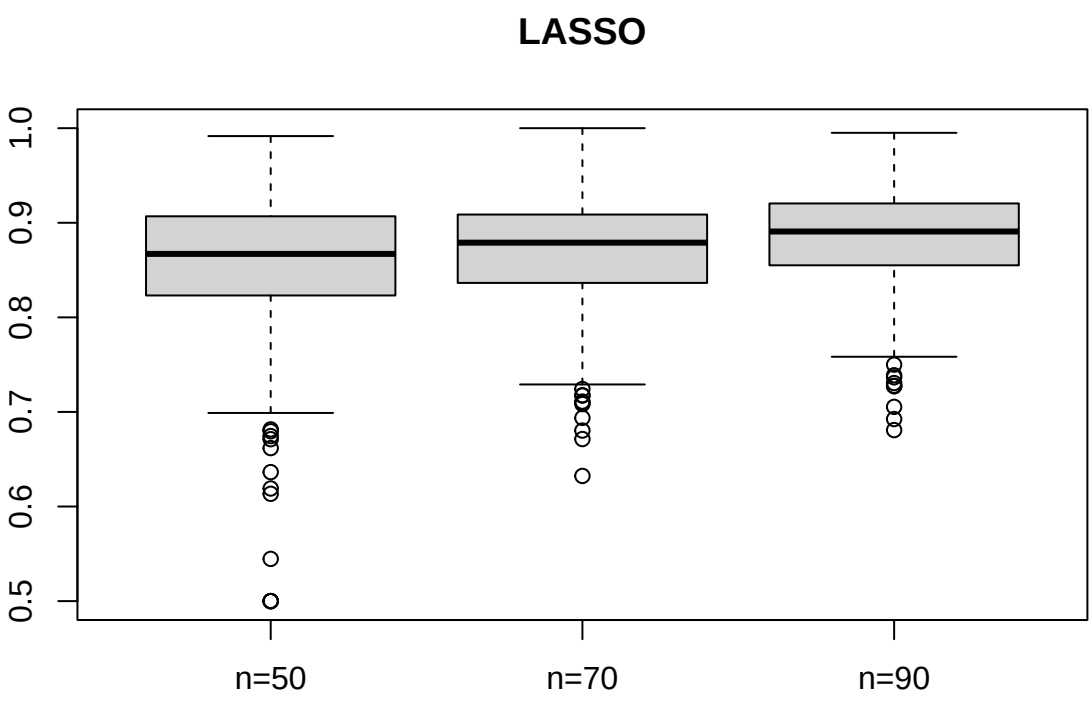
```
# se
apply(auc_supervised, 2, sd)
```

```
##  n=50,LASSO  n=70,LASSO  n=90,LASSO n=50,ALASSO n=70,ALASSO n=90,ALASSO
##  0.07197811  0.05588511  0.05184181  0.07300341  0.05871336  0.05415953
```

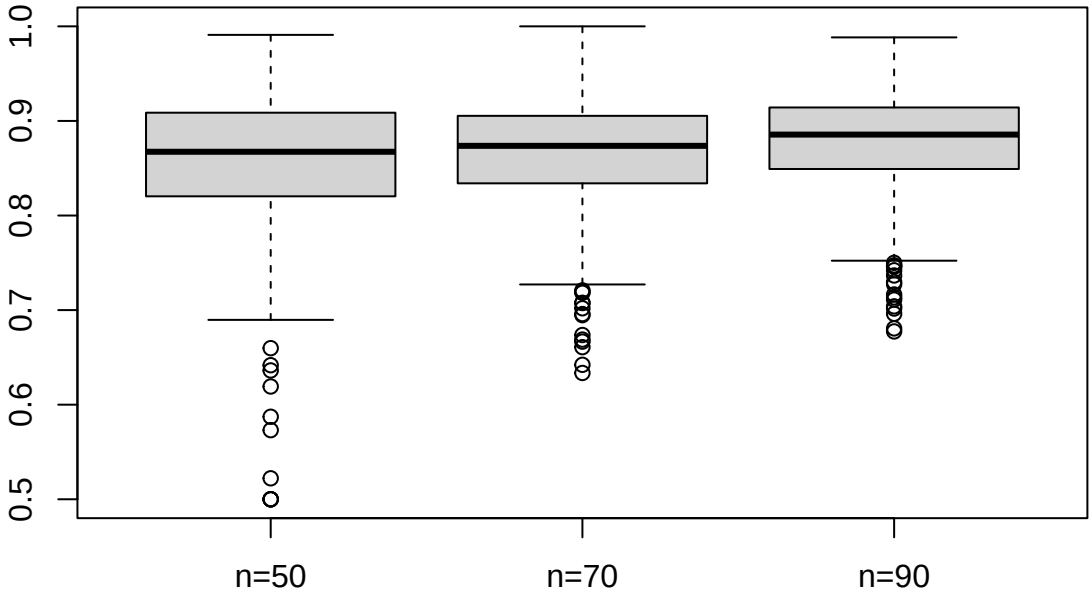
```
par(mfrow =c(1,3))
boxplot(auc_supervised[,c(1,4)], ylim = c(0.5, 1),
        names = c("LASSO", "ALASSO"), main = "n=50")
boxplot(auc_supervised[,c(1,4)], ylim = c(0.5, 1),
        names = c("LASSO", "ALASSO"), main = "n=70")
boxplot(auc_supervised[,c(1,4)], ylim = c(0.5, 1),
        names = c("LASSO", "ALASSO"), main = "n=90")
```




```
boxplot(auc_supervised[,1:3], ylim = c(0.5, 1),
        names = c("n=50", "n=70", "n=90"), main = "LASSO")
```




```
boxplot(auc_supervised[,4:6], ylim = c(0.5, 1),
        names = c("n=50", "n=70", "n=90"), main = "ALASSO")
```


ALASSO
1.0
0.9
0.8
0.7
0.6
0.5
n=50
n=70
n=90