Module 2: Supervised Learning

Jianhui Gao, Siyue Yang, and Jessica 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("../data/CAD_norm_pub.rda")
```

Prepare data for algorithm development

- Split data into training and testing set
- Training 106, Testing 75

```
ehr_data <- cbind(1:nrow(x), y, x)
colnames(ehr_data) <- c("patient_id", "label", colnames(x))
data <- PhecapData(ehr_data, "healthcare_utilization", "label", 75,
  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"),
```

```r
  surrogate, healthcare_utilization
)
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"),
    surrogate, 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"),
    surrogate, 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]
```

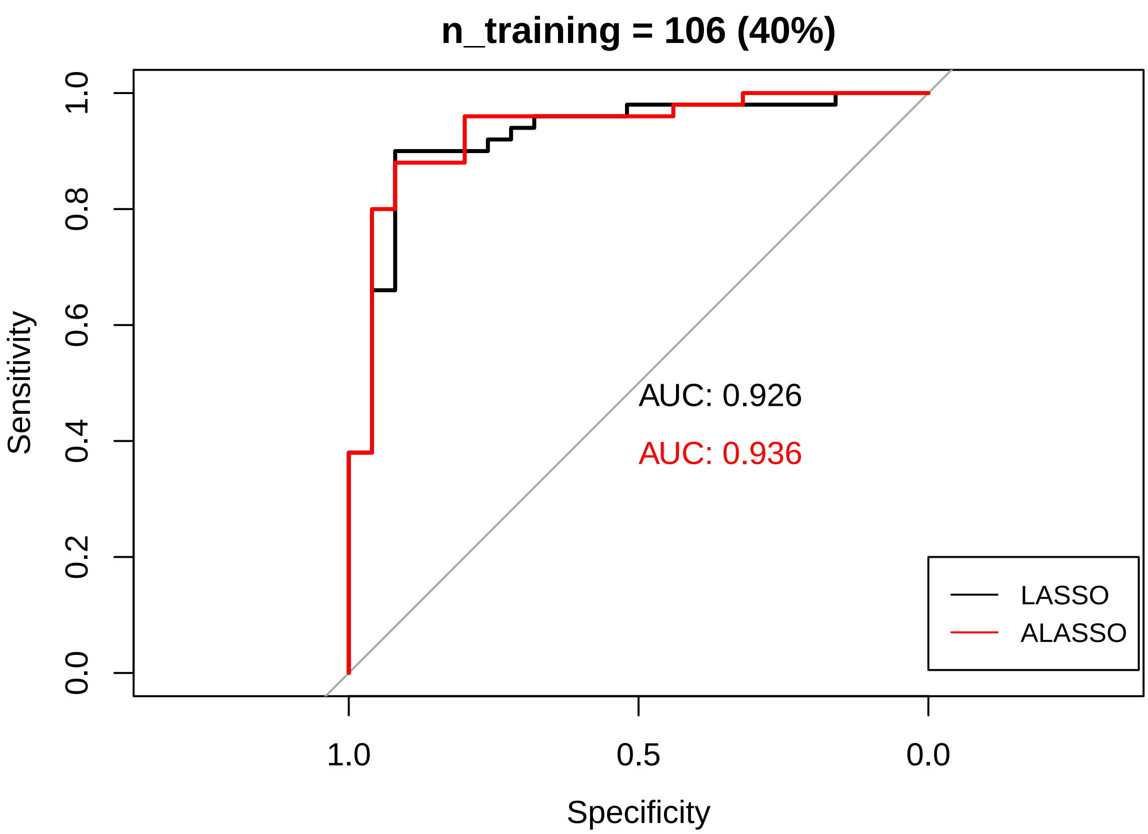
```
## [1] "NLP93"                  "NLP304"                 "surrogate"
## [4] "healthcare_utilization"
```

```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"                  "surrogate"               "healthcare_utilization"
```

```r
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 = 106 (40%)"
)
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)
```




```r
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,] 0.9438312 0.006666667 0.00 0.18050 1.0000000 0.3789314 0.3058026
##  [2,] 0.9033418 0.093333333 0.00 0.28025 1.0000000 0.4099201 0.4378051
##  [3,] 0.8628523 0.260000000 0.02 0.38000 0.9743590 0.4414414 0.5467626
##  [4,] 0.8605276 0.266666667 0.04 0.45000 0.9574468 0.4660194 0.6122449
##  [5,] 0.8582029 0.273333333 0.04 0.52000 0.9629630 0.5000000 0.6753247
##  [6,] 0.8015623 0.400000000 0.04 0.59000 0.9672131 0.5393258 0.7329193
##  [7,] 0.7449217 0.460000000 0.06 0.66000 0.9565217 0.5802469 0.7810651
##  [8,] 0.7430289 0.466666667 0.08 0.72000 0.9473684 0.6216216 0.8181818
##  [9,] 0.7411362 0.473333333 0.08 0.78000 0.9512195 0.6764706 0.8571429
## [10,] 0.6939832 0.573333333 0.08 0.84000 0.9545455 0.7419355 0.8936170
```

```r
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,] 0.9438312 0.006666667 0.00 0.18050 1.0000000 0.3789314 0.3058026
##  [2,] 0.9033418 0.093333333 0.00 0.28025 1.0000000 0.4099201 0.4378051
##  [3,] 0.8628523 0.260000000 0.02 0.38000 0.9743590 0.4414414 0.5467626
##  [4,] 0.8605276 0.266666667 0.04 0.45000 0.9574468 0.4660194 0.6122449
##  [5,] 0.8582029 0.273333333 0.04 0.52000 0.9629630 0.5000000 0.6753247
##  [6,] 0.8015623 0.400000000 0.04 0.59000 0.9672131 0.5393258 0.7329193
##  [7,] 0.7449217 0.460000000 0.06 0.66000 0.9565217 0.5802469 0.7810651
##  [8,] 0.7430289 0.466666667 0.08 0.72000 0.9473684 0.6216216 0.8181818
##  [9,] 0.7411362 0.473333333 0.08 0.78000 0.9512195 0.6764706 0.8571429
## [10,] 0.6939832 0.573333333 0.08 0.84000 0.9545455 0.7419355 0.8936170
```