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 = 1234
)

# 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"),
  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)

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

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

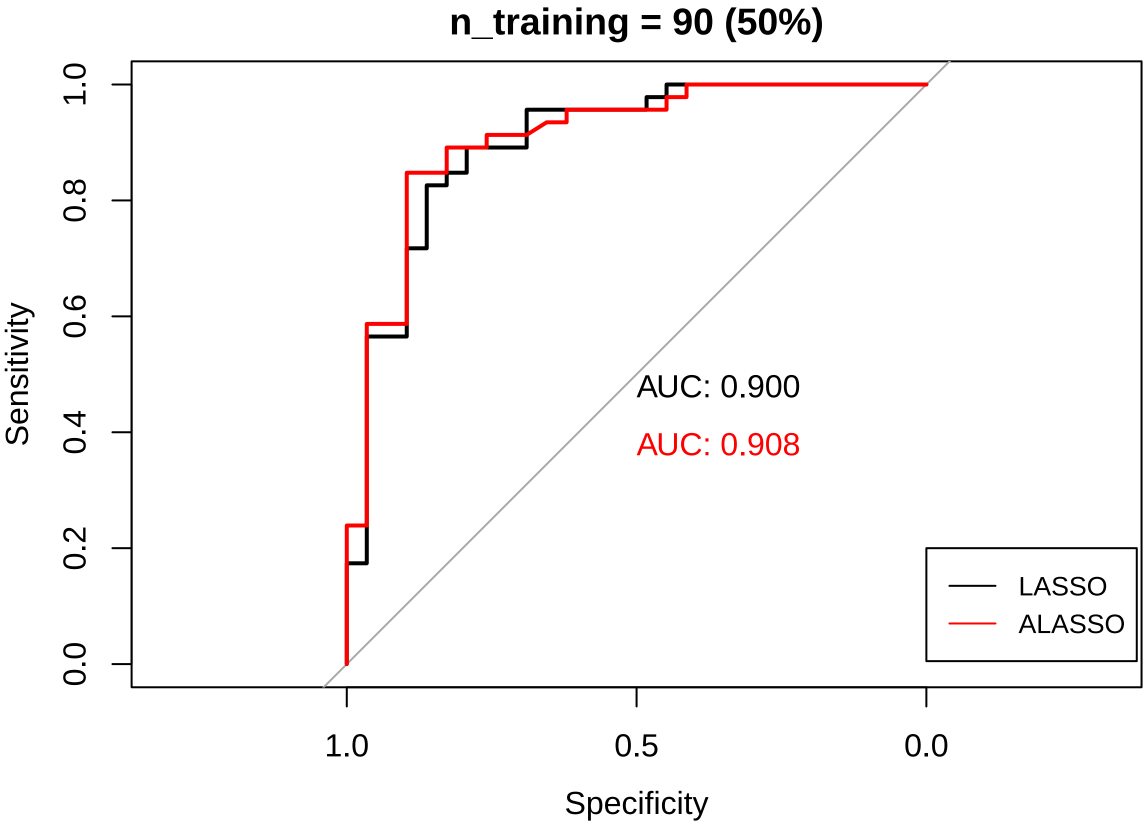
```
## [1] "NLP20"                  "NLP288"                 "NLP304"
## [4] "NLP405"                 "surrogate"              "healthcare_utilization"
```

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

```
# 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"
```

```
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,] 0.9469064 0.006666667 0.00000000 0.07729469 1.0000000 0.4059098 0.1434978
##  [2,] 0.9246943 0.080000000 0.00000000 0.13333333 1.0000000 0.4211036 0.2352941
##  [3,] 0.9080636 0.120000000 0.03448276 0.20521739 0.9042146 0.4337051 0.3345145
##  [4,] 0.9060840 0.120000000 0.03448276 0.31869565 0.9361430 0.4718571 0.4755109
##  [5,] 0.8782036 0.253333333 0.03448276 0.43217391 0.9521073 0.5173688 0.5944976
##  [6,] 0.8292790 0.346666667 0.03448276 0.54565217 0.9616858 0.5725971 0.6962552
##  [7,] 0.8202000 0.373333333 0.06896552 0.56521739 0.9285714 0.5744681 0.7027027
##  [8,] 0.8194223 0.373333333 0.06896552 0.56521739 0.9285714 0.5744681 0.7027027
##  [9,] 0.8186445 0.373333333 0.06896552 0.56521739 0.9285714 0.5744681 0.7027027
## [10,] 0.8139544 0.386666667 0.10344828 0.58195652 0.8992274 0.5748397 0.7066121
```

```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,] 0.9469064 0.006666667 0.00000000 0.07729469 1.0000000 0.4059098 0.1434978
##  [2,] 0.9246943 0.080000000 0.00000000 0.13333333 1.0000000 0.4211036 0.2352941
##  [3,] 0.9080636 0.120000000 0.03448276 0.20521739 0.9042146 0.4337051 0.3345145
##  [4,] 0.9060840 0.120000000 0.03448276 0.31869565 0.9361430 0.4718571 0.4755109
##  [5,] 0.8782036 0.253333333 0.03448276 0.43217391 0.9521073 0.5173688 0.5944976
##  [6,] 0.8292790 0.346666667 0.03448276 0.54565217 0.9616858 0.5725971 0.6962552
##  [7,] 0.8202000 0.373333333 0.06896552 0.56521739 0.9285714 0.5744681 0.7027027
##  [8,] 0.8194223 0.373333333 0.06896552 0.56521739 0.9285714 0.5744681 0.7027027
##  [9,] 0.8186445 0.373333333 0.06896552 0.56521739 0.9285714 0.5744681 0.7027027
## [10,] 0.8139544 0.386666667 0.10344828 0.58195652 0.8992274 0.5748397 0.7066121
```