Supervised Learning

Jianhui Gao

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
)
```

```
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)

```
# 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] "NLP304"                 "NLP524"                 "main_NLP"
## [4] "healthcare_utilization"
```

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

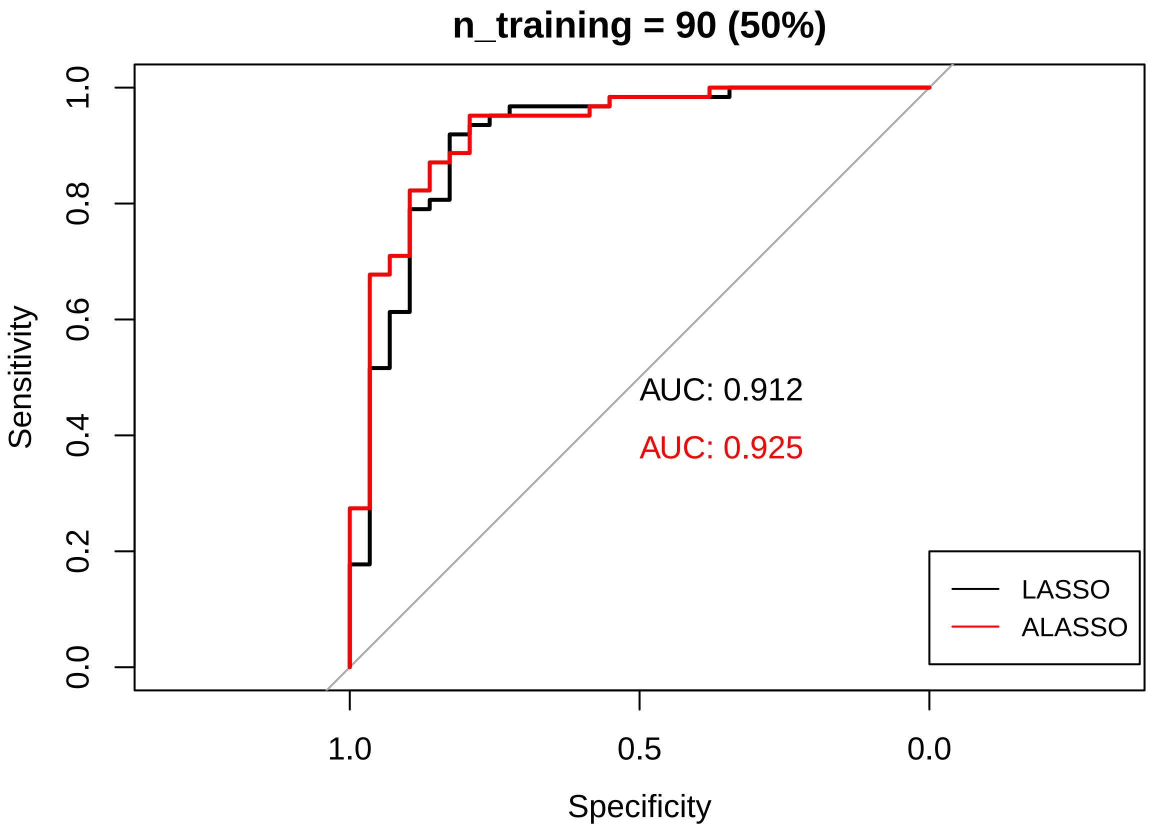
```
# Fit Adaptive LASSO
beta.alasso <- adaptive_lasso_fit(x = train_x, y = train_y,
                                  tuning = "cv", family = "binomial")
y_hat.alasso <- linear_model_predict(beta = beta.alasso, x = test_x,
                                    probability = TRUE)


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

```
## [1] "NLP304"                 "main_NLP"               "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)
```



```
FPR = 0.05
rbind(coords(roc = roc(test_y, y_hat.lasso), x = FPR, input = "fpr")[-1],
coords(roc = roc(test_y, y_hat.alasso), x = FPR, input = "fpr")[-1])
```

```
##   specificity sensitivity
## 1        0.95   0.5161290
## 2        0.95   0.6774194
```

```
FPR = 0.1
rbind(coords(roc = roc(test_y, y_hat.lasso), x = FPR, input = "fpr")[-1],
coords(roc = roc(test_y, y_hat.alasso), x = FPR, input = "fpr")[-1])
```

```
##   specificity sensitivity
## 1         0.9   0.6129032
## 2         0.9   0.7096774
```

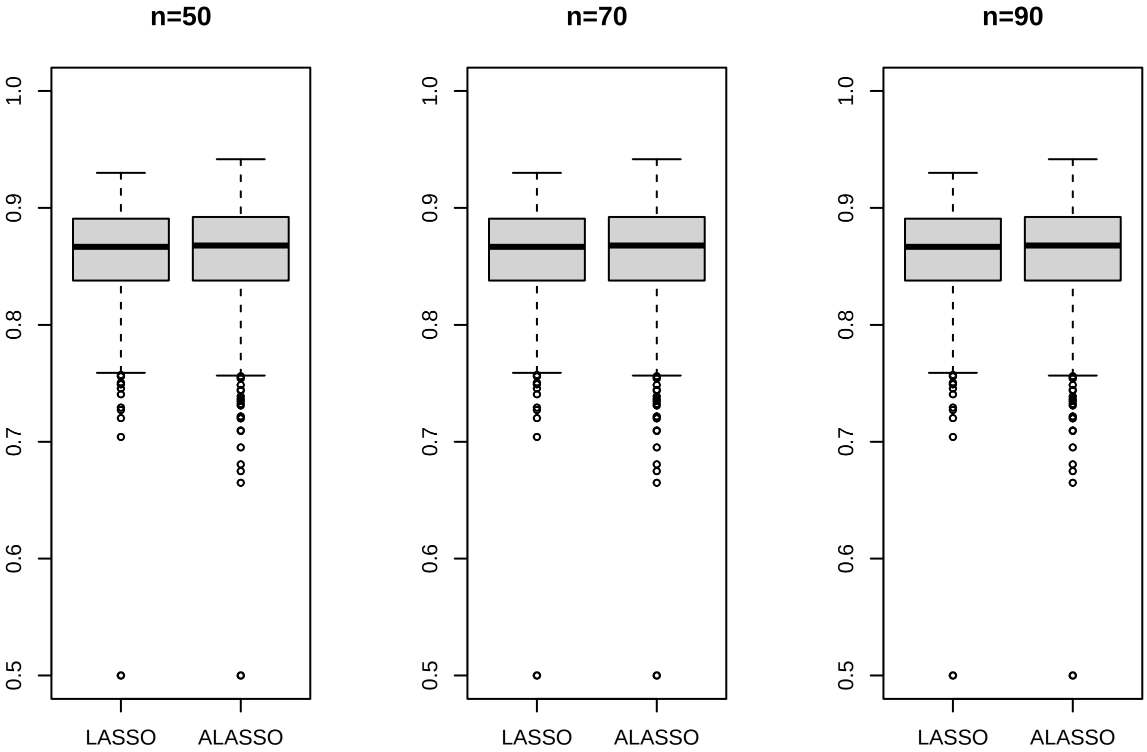
Different train size

- randomly sample traning size = 50, 70, 90
- rest as testing set
- repeat 500 times

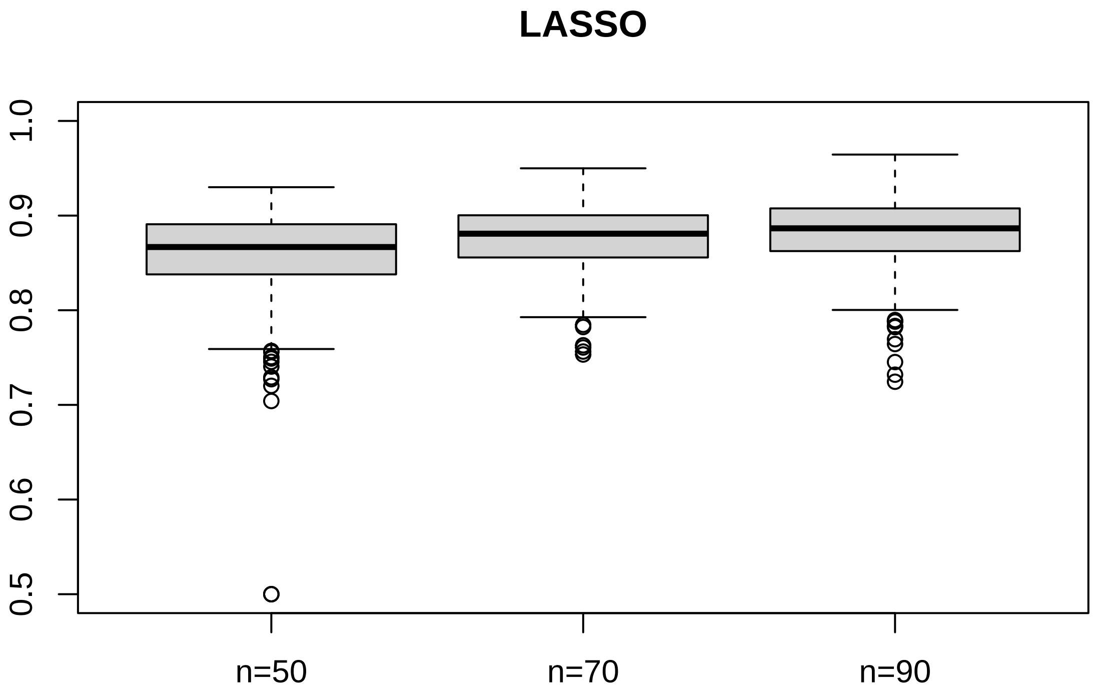
```
start<- Sys.time()
auc_supervised <- validate_supervised(dat = labeled_data, nsim = 500,
                                      n.train = c(50, 70, 90))
end <- Sys.time()
end - start
```

```
## Time difference of 2.730196 mins
```

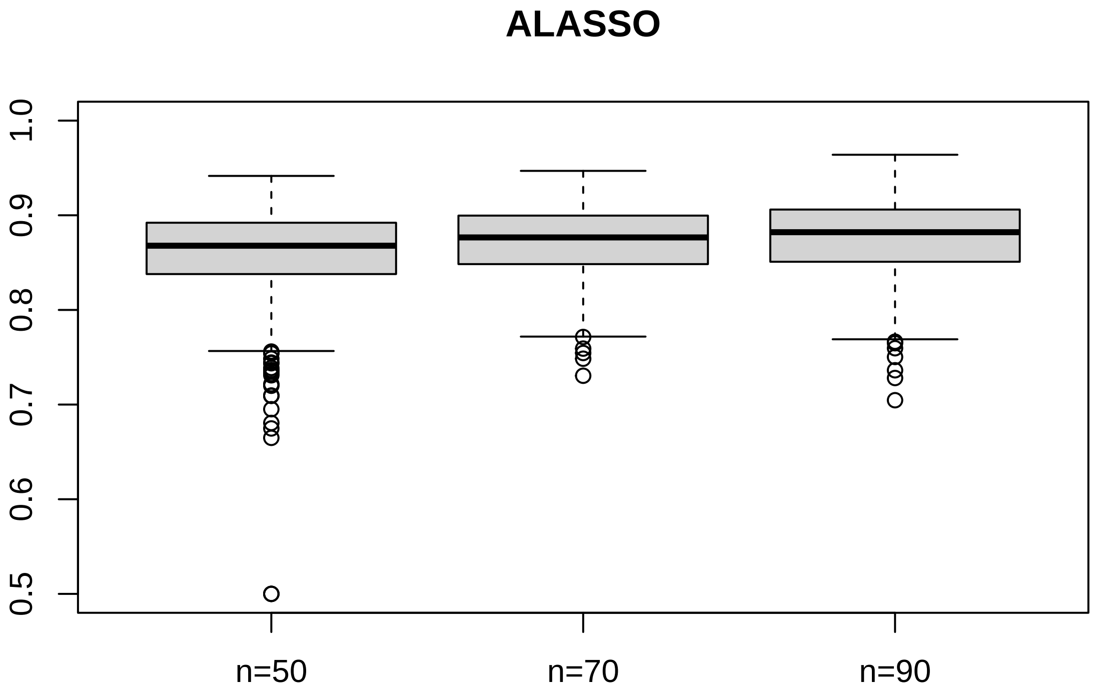
```
par(mfrow =c(1,3))
boxplot(auc_supervised[,c(1,4)], ylim = c(0.5, 1), names = c("LASSO", "ALASSO"), main = "n=50")
boxplot(auc_supervised[,c(1,4)], ylim = c(0.5, 1), names = c("LASSO", "ALASSO"), main = "n=70")
boxplot(auc_supervised[,c(1,4)], ylim = c(0.5, 1), names = c("LASSO", "ALASSO"), main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



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



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


ALASSO
n=50
n=70
n=90