Module 3: Semi-supervised learning (PheCAP)

Siyue Yang, Jianhui Gao, and Jesse Gronsbell

```
# Load the packages.
packages <- c("tidyverse", "PheCAP", "glmnet", "glmpath", "pROC")

# Check if the pacakages are missing or not.
# If missing, install automatically.
# If not missing, load the package.
package.check <- lapply(
      packages,
      FUN = function(x) {
        if (!require(x, character.only = TRUE)) {
          install.packages(x, dependencies = TRUE)
          library(x, character.only = TRUE)
        }
      }
    )
```

```
# Load helper functions.
source("../Rscripts/helper_function.R")
```

```
load('environment_pass.RData')
```

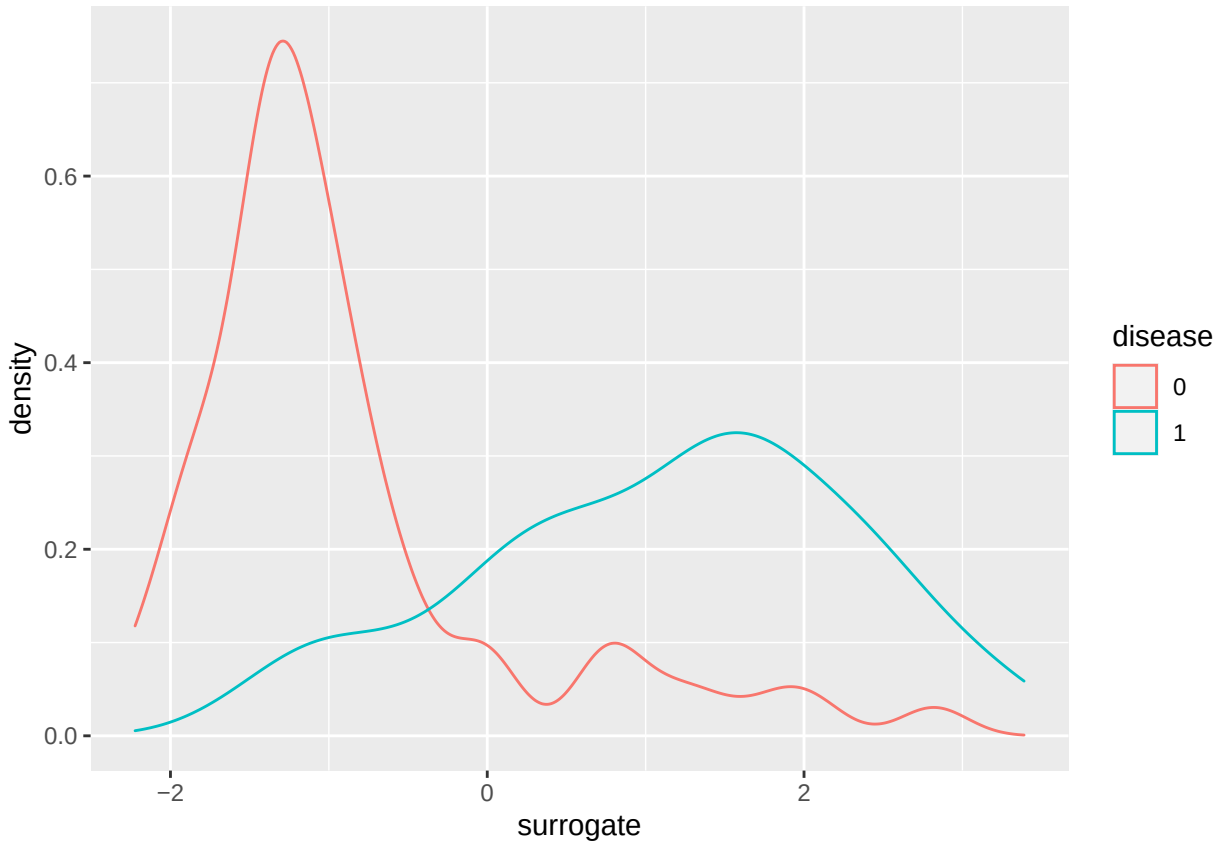
Feature selection

How to select features?

Can leverage some clinical-meaningful features that are related to Y.

e.g. Feature "surrogate" = the total number of the disease-related billing codes + disease-specific NLP mentions.

```
cbind(label = y, x) %>%
  filter(!is.na(label)) %>%
  mutate(disease = factor(label)) %>%
  ggplot(aes(x = surrogate)) +
  geom_density(aes(color = disease))
```



The more the disease-related codes, the more **likely** the patient has the disease.

```
nonmissing_index <- which(!is.na(y))
surrogate <- x$surrogate

get_auc(y[nonmissing_index], surrogate[nonmissing_index])
```

```
## [1] 0.8877745
```

We call these highly predictive features of the true disease status "surrogates".

Opportunities of using surrogate features

1. Feature selection to reduce p
2. Algorithm development with limited Y
3. Algorithm validation with limited Y

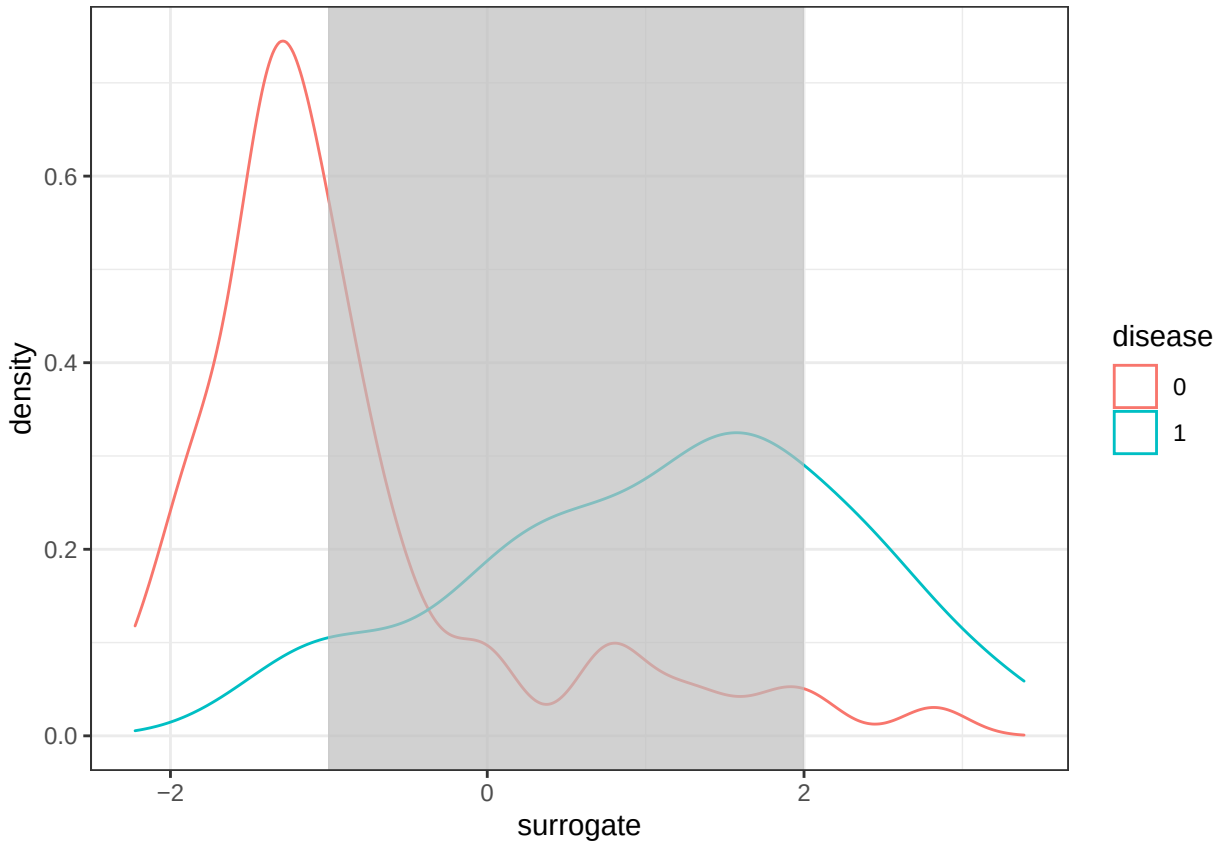
Opportunity 2 and 3 will be covered in the next module!

Feature selection method

Motivation (Extreme assumption):

- Patients with **high** main ICD or NLP mentions generally have the phenotype.
- Patients with **extremely** low counts are unlikely to have the phenotype.

```
cbind(label = y, x) %>%
  filter(!is.na(label)) %>%
  mutate(disease = factor(label)) %>%
  ggplot(aes(x = surrogate)) +
  geom_density(aes(color = disease)) +
  theme_bw() +
  annotate("rect", fill = "grey", alpha = 0.7, xmin = -1, xmax = 2,
           ymin = -Inf, ymax = Inf)
```



- Left white rect: patients not having the disease.
- Right white rect: patients having the disease.

Prepare data for feature selection

Prepare surrogates

Surrogates are available for all the patients!

```
# Prepare 3 surrogates.
surrogate <- x$surrogate

# Prepare features to be selected.
features <- data.matrix(x %>% select(starts_with("COD") | starts_with("NLP")))
```

Run surrogate-assisted feature extraction (SAFE) and show result.

```
# Truncated at 2 and -1.
SAFE <- extreme_method(surrogate, features, u_bound = 2, l_bound = -1)
SAFE_feature <- colnames(features)[SAFE$beta_select]
SAFE_feature
```

```
##  [1] "NLP56"  "NLP93"  "NLP160" "NLP161" "NLP176" "NLP231" "NLP304" "NLP306"
##  [9] "NLP309" "NLP321" "NLP349" "NLP403" "NLP434" "NLP446" "NLP456" "NLP495"
```

We select features that occur 50% among the three different surrogate-selected feature sets. This is the idea of *majority voting*.

Train phenotyping model and show the AUC on the testing set.

- Split data into training and testing set
- Training 60% (n = 106), Testing 40% (n = 75)

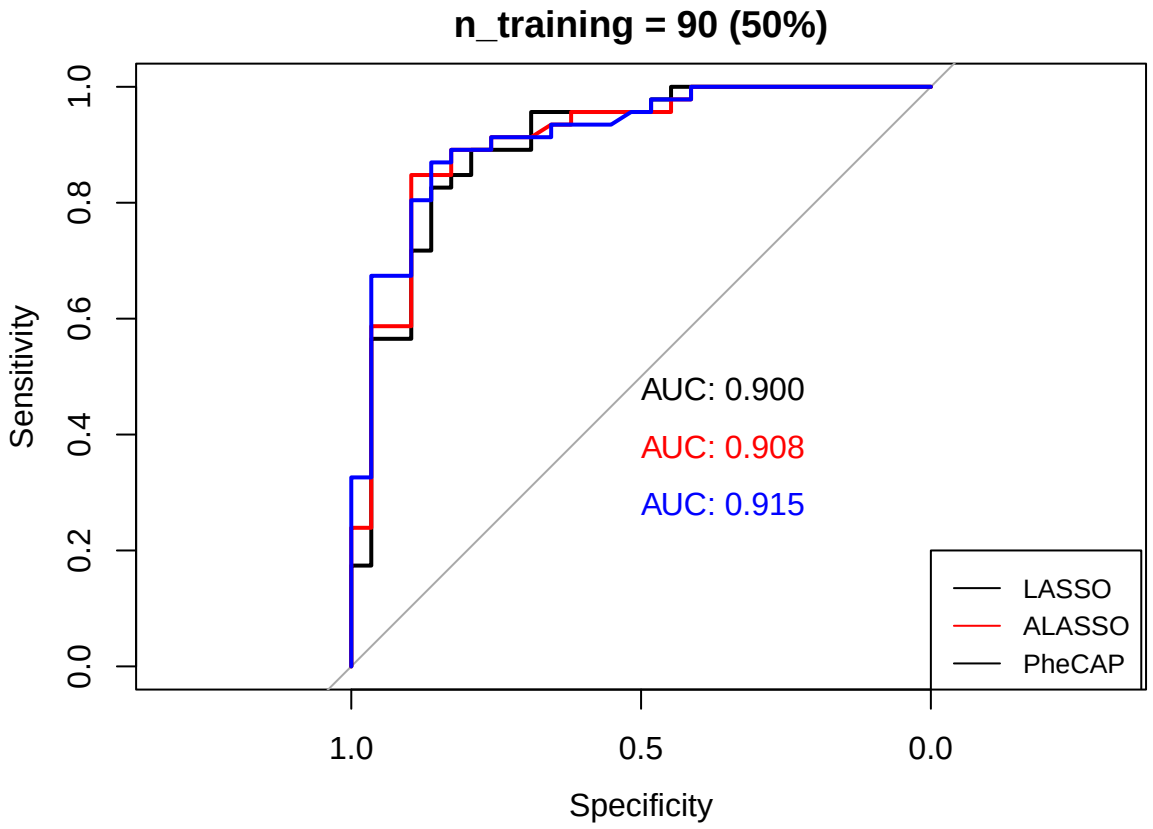
```
selected_features <- c("surrogate", "healthcare_utilization", SAFE_feature)

beta.phecap <- adaptive_lasso_fit(x = train_x[, selected_features],
                                  y = train_y,
                                  tuning = "cv", family = "binomial")

y_hat.phecap <- linear_model_predict(beta = beta.phecap,
                                     x = test_x[, selected_features],
                                     probability = TRUE)
```

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

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
)
plot(roc.phecap,
  print.auc = TRUE, col = 'blue', add = TRUE, print.auc.y = 0.3
)
legend(0, 0.2, legend = c("LASSO", "ALASSO", "PheCAP"), 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
```

```
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
```

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

```
##          cutoff    pos.rate        FPR       TPR       PPV       NPV        F1
##  [1,] 0.9943500 0.006666667 0.00000000 0.1528533 1.0000000 0.4266667 0.2651738
##  [2,] 0.9827189 0.146666667 0.00000000 0.2533288 1.0000000 0.4577969 0.4042496
##  [3,] 0.9735231 0.213333333 0.03448276 0.3539130 0.9421296 0.4851005 0.5145386
##  [4,] 0.9707201 0.213333333 0.03448276 0.4547826 0.9543796 0.5275057 0.6160188
##  [5,] 0.9527639 0.293333333 0.03448276 0.5556522 0.9623494 0.5780347 0.7045204
##  [6,] 0.9224958 0.413333333 0.03448276 0.6565217 0.9679487 0.6392694 0.7823834
##  [7,] 0.9160622 0.440000000 0.06896552 0.6739130 0.9393939 0.6428571 0.7848101
##  [8,] 0.9145942 0.440000000 0.06896552 0.6739130 0.9393939 0.6428571 0.7848101
##  [9,] 0.9131261 0.440000000 0.06896552 0.6739130 0.9393939 0.6428571 0.7848101
## [10,] 0.9105183 0.453333333 0.10344828 0.6882609 0.9134449 0.6445216 0.7850236
```

Save the data and feature selected for module 4 and model fitting.

```
save(list = ls(), file = "../module4/environment_pass.RData")
```