Module 4: Alternative approaches

2-step Semi-supervised Approach

1. Regress the surrogate on the features with penalized least square to get the direction of beta.

```r
# COD + NLP + HU.
x <- log(ehr_data %>% select(starts_with("health") |
  starts_with("COD") | starts_with("NLP")) + 1)
S <- log(ehr_data$main_ICD + ehr_data$main_NLP + 1)

# Step 1.
beta_step1 <- adaptive_lasso_fit(
  y = S[], # surrogate
  x = x[], # all X
  family = "gaussian",
  tuning = "cv"
)
```

2-step Semi-supervised Approach

1. Regress the surrogate on the features with penalized least square to get the direction of beta.
2. Regress the outcome on the linear predictor to get the intercept and multiplier for the beta.

```
# Linear predictor without intercept.
bhatx <- linear_model_predict(beta = beta_step1, x = as.matrix(x))

# Step 2.
step2 <- glm(
  train_y ~ bhatx[train_data$patient_id] + S[train_data$patient_id],
  family = "binomial"
)
beta_step2 <- coef(step2)
beta_step2
```

```
##                 (Intercept) bhatx[train_data$patient_id]
##                  -1.9461028                    0.7057629
##    S[train_data$patient_id]
##                   0.5988575
```

```
# Recover beta.
beta <- beta_step2[2] * beta_step1
```

Compare selected features

```r
# LASSO.
names(beta_lasso[!beta_lasso == 0])[-1]
```

```
##  [1] "COD2"     "COD10"    "NLP1"     "NLP17"    "NLP56"    "NLP82"   
##  [7] "NLP93"    "NLP104"   "NLP118"   "NLP130"   "NLP144"   "NLP164"  
## [13] "NLP172"   "NLP193"   "NLP199"   "NLP222"   "NLP231"   "NLP265"  
## [19] "NLP274"   "NLP280"   "NLP297"   "NLP299"   "NLP346"   "NLP362"  
## [25] "NLP375"   "NLP382"   "NLP396"   "NLP401"   "NLP409"   "NLP435"  
## [31] "NLP451"   "NLP462"   "NLP488"   "NLP533"   "NLP536"   "NLP552"  
## [37] "NLP568"   "main_NLP"
```

```r
# ALASSO.
names(beta_alasso[!beta_alasso == 0])[-1]
```

```
##  [1] "NLP56"    "NLP93"    "NLP104"   "NLP118"   "NLP222"   "NLP231"  
##  [7] "NLP265"   "NLP280"   "NLP297"   "NLP299"   "NLP409"   "NLP536"  
## [13] "main_NLP"
```

Compare selected features

```r
# PheCAP.
feature_selected
```

```
## Feature(s) selected by surrogate-assisted feature extraction (SAFE)
## [1] "main_ICD" "main_NLP" "NLP56"    "NLP93"    "NLP274"   "NLP306"
```

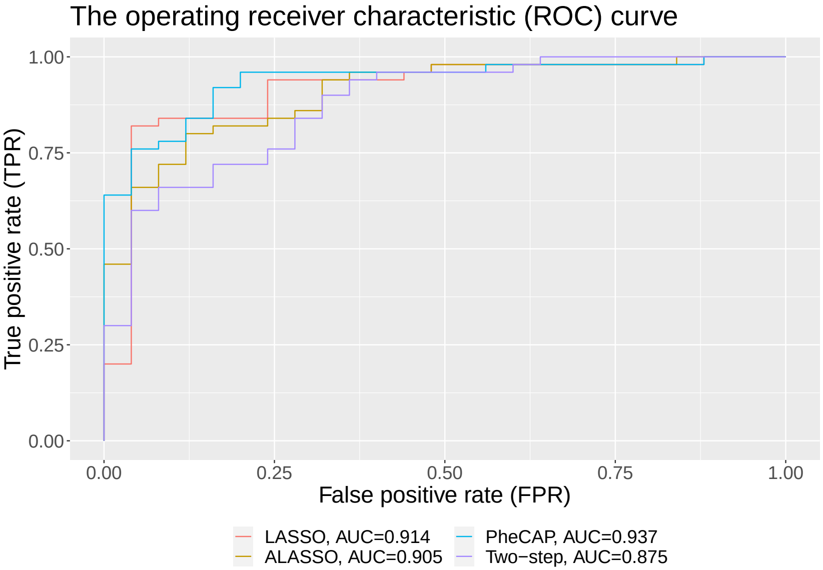
```r
# Two Step.
names(beta[!beta == 0])[-1]
```

```
##   [1] "COD6"   "COD8"   "COD10"  "NLP7"   "NLP14"  "NLP21"  "NLP24"  "NLP28" 
##   [9] "NLP31"  "NLP33"  "NLP44"  "NLP50"  "NLP56"  "NLP59"  "NLP61"  "NLP62" 
##  [17] "NLP66"  "NLP68"  "NLP70"  "NLP73"  "NLP74"  "NLP76"  "NLP81"  "NLP89" 
##  [25] "NLP92"  "NLP93"  "NLP95"  "NLP98"  "NLP102" "NLP104" "NLP108" "NLP110"
##  [33] "NLP116" "NLP127" "NLP130" "NLP146" "NLP160" "NLP161" "NLP172" "NLP176"
##  [41] "NLP178" "NLP179" "NLP183" "NLP189" "NLP190" "NLP192" "NLP199" "NLP202"
##  [49] "NLP203" "NLP206" "NLP215" "NLP225" "NLP231" "NLP232" "NLP243" "NLP246"
##  [57] "NLP250" "NLP253" "NLP256" "NLP288" "NLP294" "NLP295" "NLP299" "NLP302"
##  [65] "NLP304" "NLP306" "NLP309" "NLP318" "NLP321" "NLP326" "NLP336" "NLP338"
##  [73] "NLP342" "NLP343" "NLP347" "NLP349" "NLP350" "NLP351" "NLP357" "NLP359"
##  [81] "NLP361" "NLP363" "NLP365" "NLP369" "NLP380" "NLP387" "NLP393" "NLP395"
##  [89] "NLP403" "NLP405" "NLP407" "NLP417" "NLP431" "NLP434" "NLP437" "NLP440"
##  [97] "NLP446" "NLP451" "NLP456" "NLP463" "NLP465" "NLP468" "NLP473" "NLP482"
## [105] "NLP483" "NLP487" "NLP490" "NLP495" "NLP500" "NLP507" "NLP523" "NLP529"
## [113] "NLP534" "NLP536" "NLP539" "NLP541" "NLP544" "NLP554" "NLP560" "NLP564"
## [121] "NLP568" "NLP572"
```

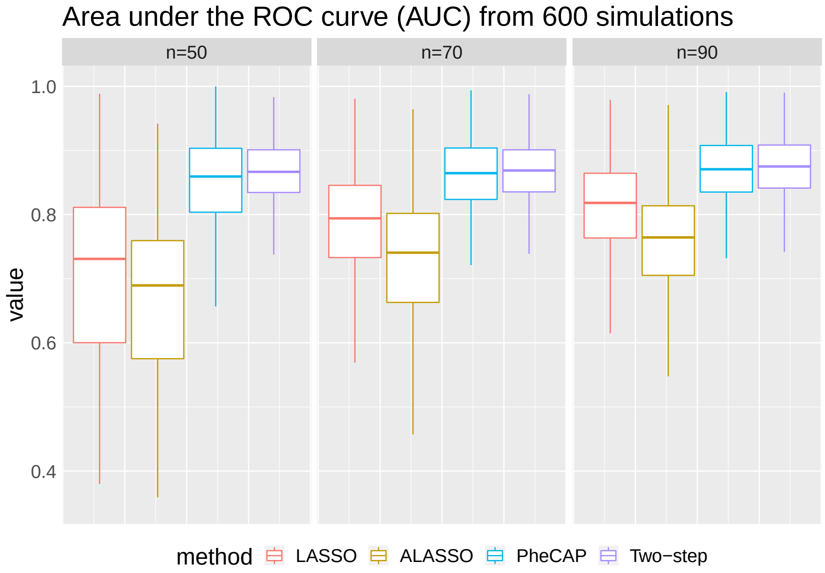
ROC

```
mu <- beta_step2[1] +
  as.numeric(as.matrix(x[test_data$patient_id, ])
  %*% beta[-1]) +
  as.numeric(beta_step2[3] %*% S[test_data$patient_id])

# Expit.
y_hat_twostep <- plogis(mu)

roc_twostep <- roc(test_y, y_hat_twostep)
```

ROC


The operating receiver characteristic (ROC) curve
True positive rate (TPR)
1.00
0.75
0.50
0.25
0.00
0.00
0.25
0.50
0.75
1.00
False positive rate (FPR)
LASSO, AUC=0.914
ALASSO, AUC=0.905
PheCAP, AUC=0.937
Two−step, AUC=0.875

Model Evaluation



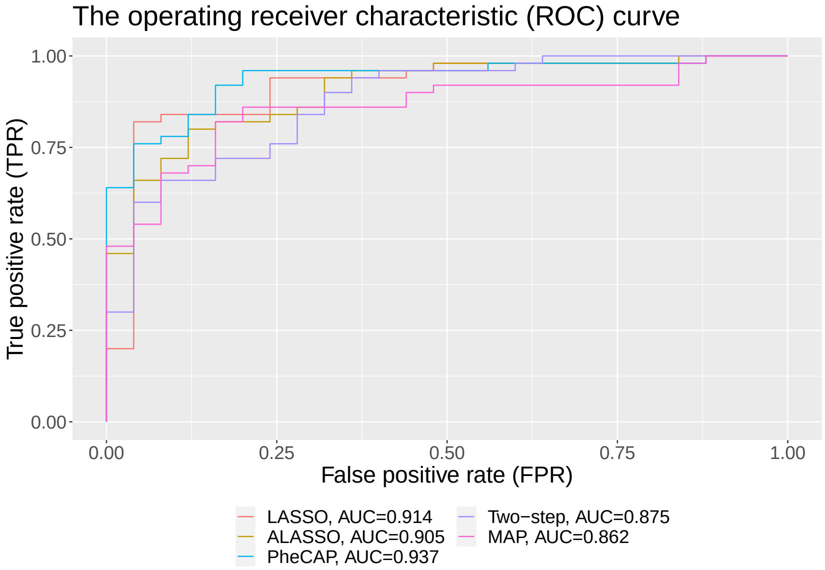
MAP

```r
# Use un-transformed data; MAP requires sparse matrix.
# Create sparse matrix for surrogates.
data_fit <- sparsify(
  PheCAP::ehr_data %>%
  select(main_ICD, main_NLP) %>%
  rename(ICD = main_ICD) %>% data.table()
)

# Create sparse matrix for HU.
note <- Matrix(
  PheCAP::ehr_data$healthcare_utilization,
  ncol = 1, sparse = TRUE
)
model_map <- MAP(mat = data_fit, note = note, full.output = TRUE)
```

```
## #######################
## MAP only considers pateints who have note count data and
##         at least one nonmissing variable!
## ####
## Here is a summary of the input data:
## Total number of patients: 10000
##   ICD main_NLP note  Freq
## 1 YES      YES  YES 10000
## ####
```

```r
y_hat_map <- model_map$scores[data$validation_set]
roc_map <- roc(test_y, y_hat_map)
```

ROC


The operating receiver characteristic (ROC) curve
True positive rate (TPR)
0.00
0.25
0.50
0.75
1.00
False positive rate (FPR)
0.00
0.25
0.50
0.75
1.00
LASSO, AUC=0.914
ALASSO, AUC=0.905
PheCAP, AUC=0.937
Two−step, AUC=0.875
MAP, AUC=0.862