

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.

```
x <- log(ehr_data %>% select(starts_with("health") |
  starts_with("COD") | starts_with("NLP")) + 1) # COD + NLP
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 = "ic"
)
```

2-step Semi-supervised Approach

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

```
# 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$patient_id]
##                -1.9395295                        0.6361248
##                S[train_data$patient_id]
##                0.6534730

# recover beta
beta <- beta_step2[2] * beta_step1
```

Compare selected features

```
# 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"
```

```
# ALASSO
```

```
names(beta_lasso[!beta_lasso == 0])[-1]
```

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

```
# PheCAP
```

```
feature_selected
```

```
## Feature(s) selected by surrogate-assisted feature extraction (SAFE)
```

```
## [1] "main_ICD" "main_NLP" "NLP56"     "NLP93"     "NLP274"    "NLP306"
```

```
# Two Step
```

```
names(beta[!beta == 0])[-1]
```

```
## [1] "COD10"     "NLP6"      "NLP14"     "NLP24"     "NLP31"     "NLP44"     "NLP56"     "NLP59"
## [9] "NLP61"     "NLP68"     "NLP73"     "NLP74"     "NLP93"     "NLP127"    "NLP130"    "NLP160"
## [17] "NLP161"    "NLP172"    "NLP176"    "NLP193"    "NLP199"    "NLP202"    "NLP215"    "NLP225"
## [25] "NLP231"    "NLP243"    "NLP294"    "NLP295"    "NLP302"    "NLP304"    "NLP306"    "NLP309"
## [33] "NLP321"    "NLP349"    "NLP350"    "NLP361"    "NLP403"    "NLP434"    "NLP446"    "NLP451"
## [41] "NLP456"    "NLP463"    "NLP465"    "NLP482"    "NLP495"    "NLP507"    "NLP536"    "NLP539"
## [49] "NLP541"    "NLP544"    "NLP560"    "NLP564"
```

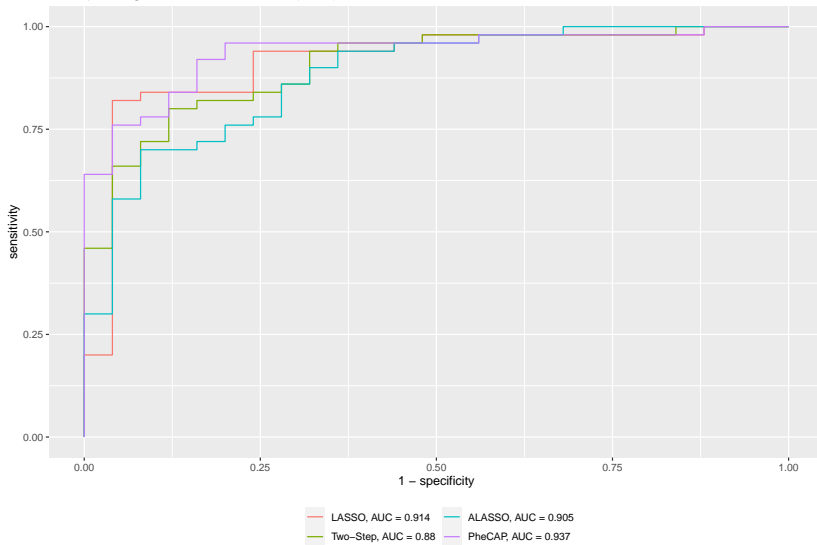
ROC

```
# mu
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)
```

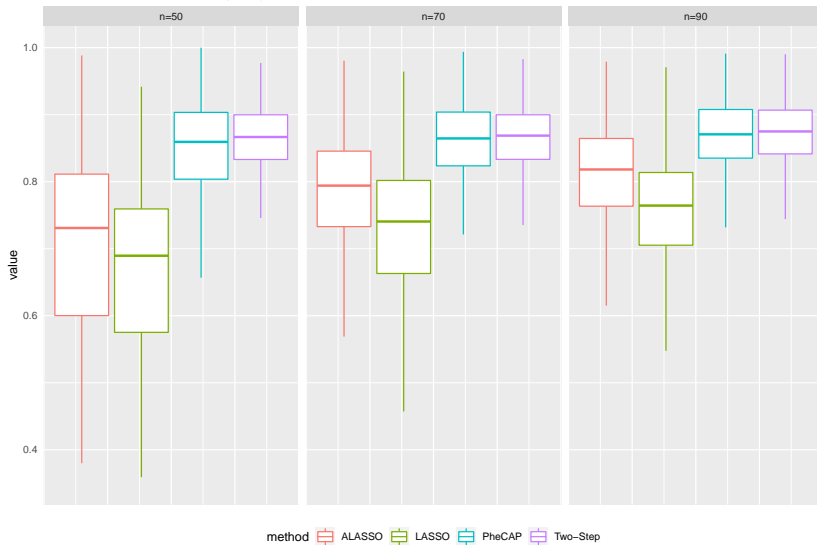
ROC

The operating receiver characteristic (ROC) curve



Model Evaluation

Area under the ROC curve (AUC) from 600 simulations



MAP

```
# Use untransformed 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 patients 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
## ####

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


ROC

The operating receiver characteristic (ROC) curve

