Penalized Survival Analysis for Burn Patients

Shannon Chen

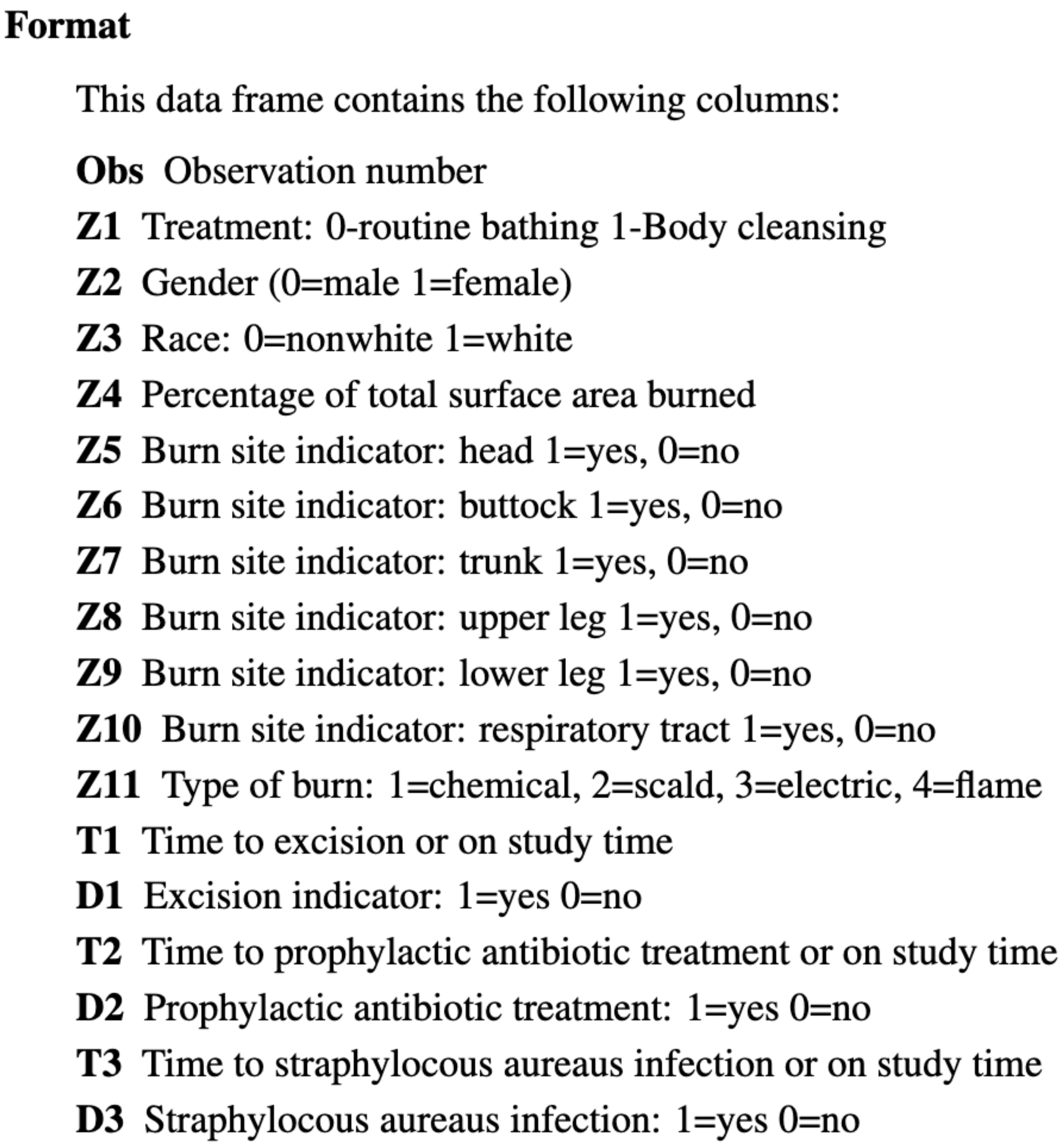
Burns are some of the most painful and terrifying injuries a person can experience. To make things worse, accidents that cause them can't be completely prevented. All it takes is awful luck and a person's life can change in an instant. Despite this grim reality we're not completely helpless since we can improve our medical sophistication to treat affected patients. Studying the recovery process is vital to improving our ability to deal with these life altering events.

The "burn" dataset from the "KMsurv" package in R has 154 rows and 18 columns. Each row represents a patient, and the attributes provide information on the patient's race, gender, percent of body burned, and the treatment process and complications. Categorical variables describe whether the patient underwent routine bathing or body cleansing, whether the patient's specific body parts were burned, what type of burn the patient has, whether the patient was given prophylactic antibiotic treatment, if the patient's wounds needed excision, and if the patient was infected with straphylocous aureaus during recovery. Continuous variables include percentage of the patient's body that was burnt, the time in days until the patient undergoes excision, is treated with antibiotics, and is infected with Straphylocous aureaus.

Medical literature has discussed the importance of early excision in the treatment of patients, since early excision can prevent infection and allow for easier wound healing. Whether the patient undergoes excision during their treatment is highly significant to the final outcome of the patient, and examining the excision process can help determine what the patient's quality of life can be after they fully recover.

Survival analysis is most appropriate for modeling the excision outcome, and due to the high number of predictors, using a multivariate approach could allow for irrelevent covariates to be eliminated while preserving significant ones. In this project, I explore different methods of high dimensional cox modelling.

```r
## Add the names of all the packages that you used in the pipeline to list.of.packages
list.of.packages <- c("survival", "KMsurv", "ggplot2", "stats", "MASS", "hdnom")

## Automatically install any necessary packages that have not already been installed
new.packages <- list.of.packages[!(list.of.packages %in% installed.packages()[,"Package"])]
if(length(new.packages))
  install.packages(new.packages)

library("survival")
library("KMsurv")
library("stats")
library("MASS")
library("hdnom")

data("burn")
burns <- burn[, -1]
burns_ex <- as.matrix(burns[, -c(12,13)]) #Removed excision indicator
burns_straph <- burns[, -c(14, 15)] #Removed straph indicator
burns_antibiotic <- burns[, -c(16, 17)] #Removed antibiotic indicator

set.seed(123)
smp_size <- floor(0.9 * nrow(burns_ex))
train_ind <- sample(seq_len(nrow(burns_ex)), size = smp_size)

ex_train <- as.matrix(burns_ex[train_ind, ])
ex_test <- as.matrix(burns_ex[-train_ind, ])
```



Format

This data frame contains the following columns:

Obs Observation number

Z1 Treatment: 0-routine bathing 1-Body cleansing

Z2 Gender (0=male 1=female)

Z3 Race: 0=nonwhite 1=white

Z4 Percentage of total surface area burned

Z5 Burn site indicator: head 1=yes, 0=no

Z6 Burn site indicator: buttock 1=yes, 0=no

Z7 Burn site indicator: trunk 1=yes, 0=no

Z8 Burn site indicator: upper leg 1=yes, 0=no

Z9 Burn site indicator: lower leg 1=yes, 0=no

Z10 Burn site indicator: respiratory tract 1=yes, 0=no

Z11 Type of burn: 1=chemical, 2=scald, 3=electric, 4=flame

T1 Time to excision or on study time

D1 Excision indicator: 1=yes 0=no

T2 Time to prophylactic antibiotic treatment or on study time

D2 Prophylactic antibiotic treatment: 1=yes 0=no

T3 Time to straphylocous aureaus infection or on study time

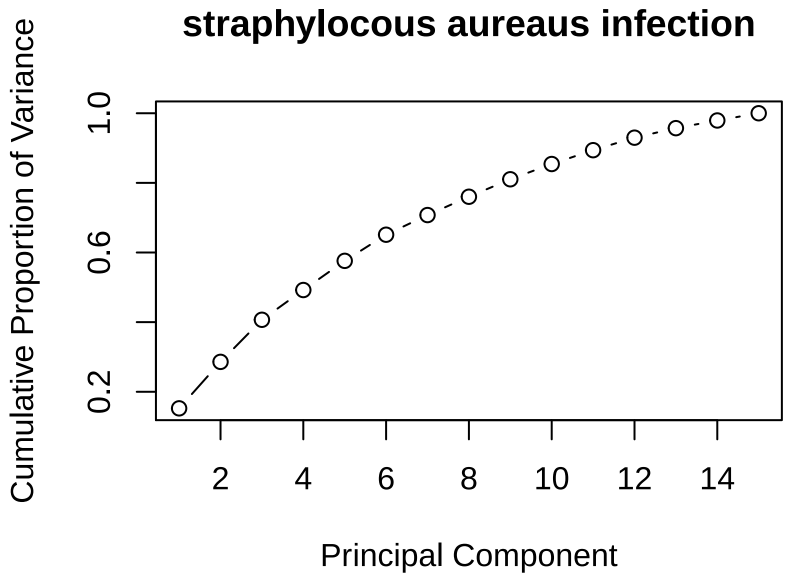
D3 Straphylocous aureaus infection: 1=yes 0=no

Figure 1: Burn Dataset Variables

```
ex_time_train <- burns[train_ind,12] #T1 is the time to excision event
ex_time_test <- burns[-train_ind,12]

excision_train <- burns[train_ind,13] #D1 is indicator for excision event
excision_test <- burns[-train_ind,13]
surv_train <- Surv(ex_time_train, excision_train)
```

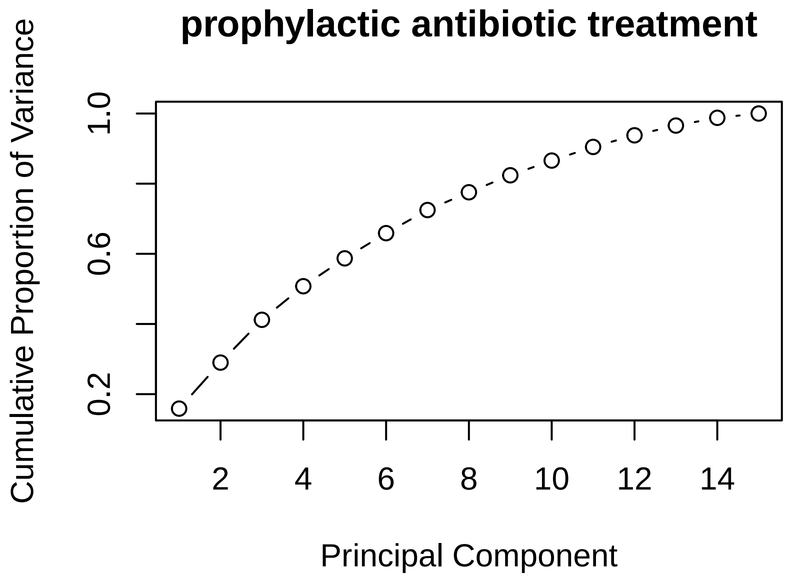
Principal Components Analysis for Variable Selection

```
straph_pr <- prcomp(burns_straph, scale. = TRUE)
excision_pr <- prcomp(burns_ex, scale. = TRUE)
antibiotic_pr <- prcomp(burns_antibiotic, scale. = TRUE)

prcompvariance <- function(prcomp_df, title){
  pr_stdev <- prcomp_df$sdev
  pr_var <- pr_stdev^2
  var_prop <- pr_var/sum(pr_var) #Proportional Variance
  cumplot <- plot(cumsum(var_prop), xlab = "Principal Component",
                  ylab = "Cumulative Proportion of Variance",
                  main = title, type = "b")
  return(cumplot)
}

prcompvariance(straph_pr, "straphylocous aureaus infection")
```



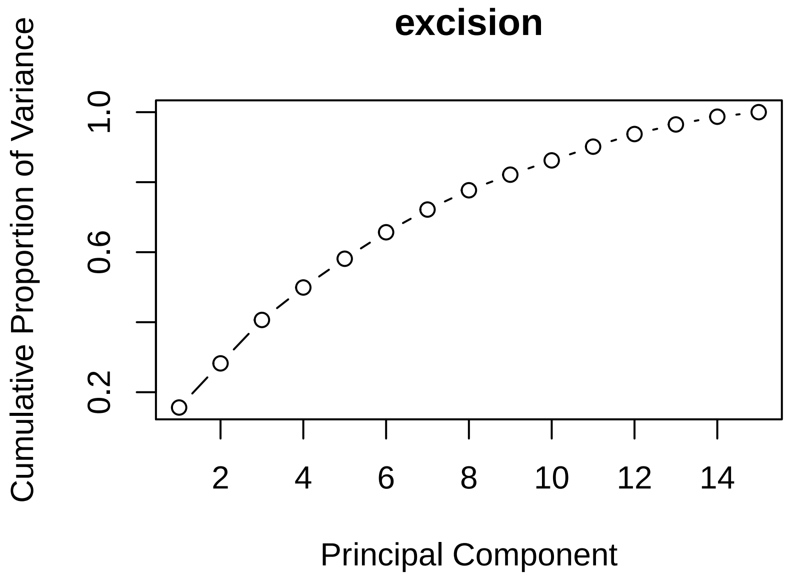
```
## NULL
```

```
prcompvariance(excision_pr, "prophylactic antibiotic treatment")
```



```
## NULL
```

```
prcompvariance(antibiotic_pr, "excision")
```



```
## NULL
```

My initial idea was to use principal components for variable selection. However, as shown, the cumulative proportion of variance graphs for the different categories (straphylocous infection, antibiotic treatment, and excision) are not very informative. There doesn't appear to be an optimal cutoff of principal components. Thus, another approach would be more appropriate since this dataset contains a mix of categorical (mostly binary) and continuous data.

Penalized Cox Model with hdnom

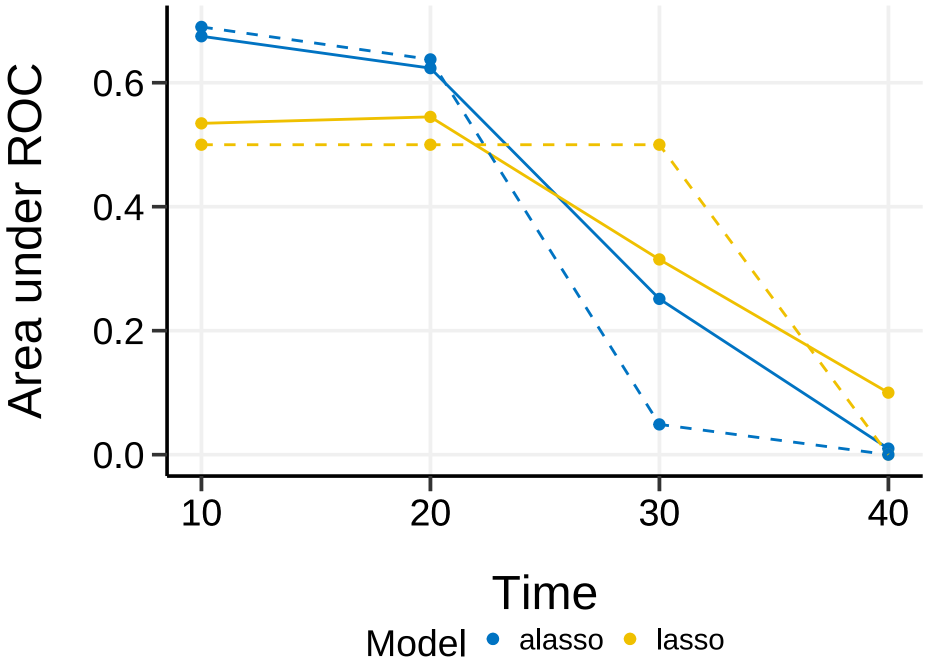
```
set.seed(124)

ex_time <- burn$T1
ex_event <- burn$D1
ex_surv <- survival::Surv(ex_time, ex_event)

excision_aenet <- fit_aenet(ex_train, surv_train, nfolds=5,  rule= "lambda.1se", seed = c(5, 7))

#lambda.1se gives the most regularized model such that
#error is within one standard error of the minimum.

excision_mcp <- fit_mcp(ex_train, surv_train, nfolds=5, seed = c(5, 7))


#For the lasso, with the train and test split the sample size wasn't enough,
#causing it to form a null model.

excision_alasso <- fit_alasso(burns_ex, ex_surv, nfolds=5, rule = "lambda.1se", seed = c(5, 7))
excision_lasso <- fit_lasso(burns_ex, ex_surv, nfolds=5, rule = "lambda.1se", seed = c(5, 7))
```

Functions in the "hdnom" package automatically tune lambda parameters for penalization. The optimal lambda values are selected through 5-fold cross validation. The different model types used to fit the data include adaptive lasso, lasso, adaptive elastic net, and minimax concave penalty (mcp). The adaptive elastic net model may be more equipped to handle collinearity between variables, and mcp is continuous and nearly unbiased.

Lasso Internal Validation Comparison

```
lasso_comp <- compare_by_validate(
  burns_ex, ex_time, ex_event,
  model.type = c("lasso", "alasso"),
  method = "cv", nfolds = 5, tauc.type = "UNO",
  tauc.time = seq(0.25, 1, 0.25) * 40,
  seed = c(5, 7), trace = FALSE
)

plot(lasso_comp)
```

```
##                  10        20        30  40
## Mean      0.5344713 0.5448920 0.3149183 0.1
## Min      0.4799052 0.3170477 0.0000000 0.0
## 0.25 Qt. 0.5000000 0.5000000 0.0000000 0.0
## Median   0.5000000 0.5000000 0.5000000 0.0
## 0.75 Qt. 0.5707414 0.6687951 0.5000000 0.0
## Max      0.6217101 0.7386175 0.5745913 0.5
##                 10        20         30          40
## Mean     0.6749747 0.6235238 0.25129199 0.009758742
## Min      0.4999505 0.2121161 0.00000000 0.000000000
## 0.25 Qt. 0.5629601 0.5807584 0.00000000 0.000000000
## Median   0.6900241 0.6376160 0.04879371 0.000000000
## 0.75 Qt. 0.7835752 0.7701033 0.30443289 0.000000000
## Max      0.8383636 0.9170249 0.90323333 0.048793710
```



```
#dashed line: median of AUC
#solid line: mean of AUC
```

To compare the performance of the adaptive lasso and regular lasso models, “hdnom” allows for internal validation by evaluation of the AUC over time. The models were used to predict the excision outcomes of patients over 40 days, with intervals every 10 days.

The graph above shows that the performance of the adaptive lasso is better than the regular lasso until around 27 days. The stability (how close together the median and mean of the AUC values are) of both models declines as well at around the same time, which is past 20 days.

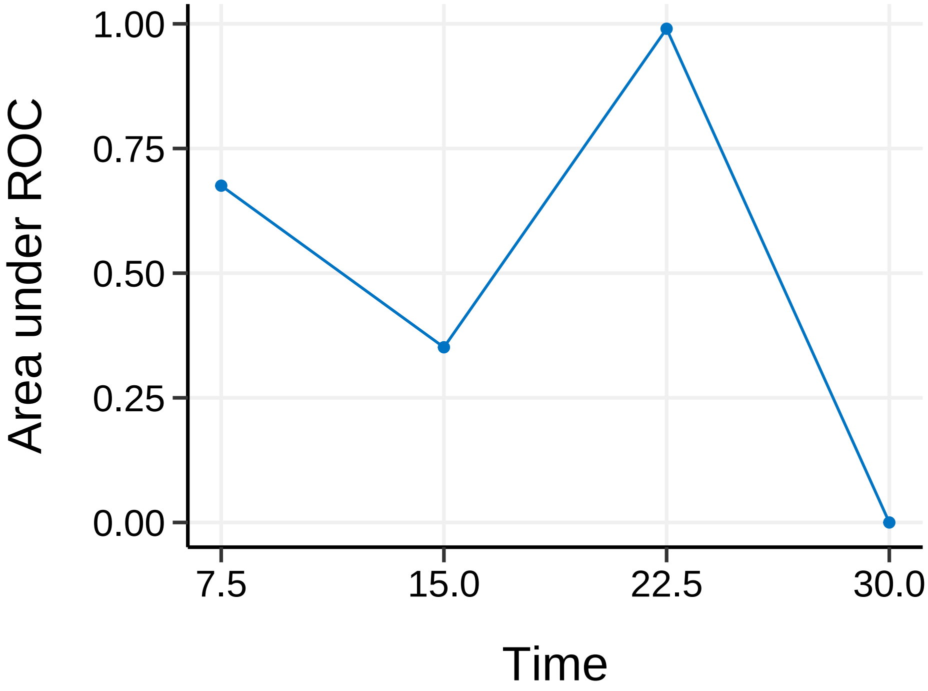
External Validation

For adaptive elastic net and mcp models, the data was split into a 9:1 ratio for training and test. The models were externally validated (although ideally, external validation would be using data collected in similar but separated studies) with the test data. The AUC is measured over 30 days, with quarter intervals in between.

External Validation of Adaptive Elastic Net

```r
set.seed(124)
aenet_val_ex <- validate_external(excision_aenet, ex_train, ex_time_train, excision_train,
                                  ex_test, ex_time_test, excision_test, tauc.type = "UNO",
                                  tauc.time = seq(0.25, 1, 0.25) * 30)
plot(aenet_val_ex, main = "Adaptive Elastic Net AUC over Time")
```

```
##           7.5        15      22.5 30
## AUC 0.6754109 0.3513219 0.9901526  0
```



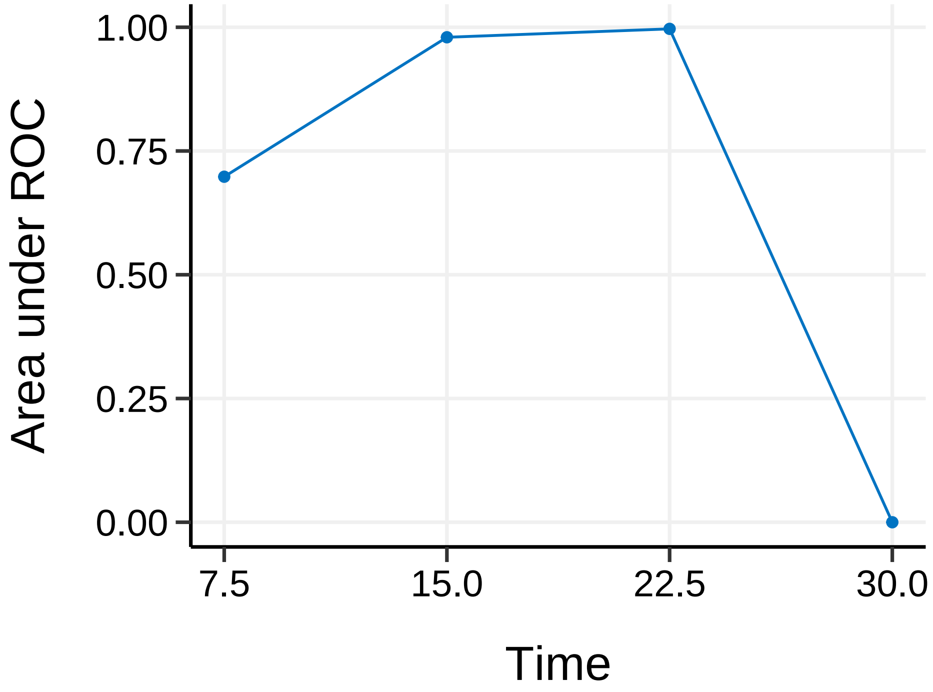
```r
summary(aenet_val_ex)
```

```
## Time-Dependent AUC Summary at Evaluation Time Points
##           7.5        15      22.5 30
## AUC 0.6754109 0.3513219 0.9901526  0
```

External Validation of MCP

```r
mcp_val_ex <- validate_external(excision_mcp, ex_train, ex_time_train, excision_train,
                                  ex_test, ex_time_test, excision_test, tauc.type = "UNO",
                                  tauc.time = seq(0.25, 1, 0.25) * 30)
plot(mcp_val_ex)
```

```
##           7.5        15      22.5 30
## AUC 0.6980054 0.9797495 0.9967175  0
```



```r
summary(mcp_val_ex)
```

```
## Time-Dependent AUC Summary at Evaluation Time Points
##           7.5        15      22.5 30
## AUC 0.6980054 0.9797495 0.9967175  0
```

As shown by the AUC values, the mcp model surpasses the adaptive elastic net in all four interval points. While comparatively a better model, it should be acknowledged that the test data has only 16 rows. The reason for this is that the dataset itself is relatively small, and so if a 3:1 ratio were used for the training and test split, null models (where coefficients of every variable is penalized to 0) would be produced. Thus, in an ideal situation, the models would be tested over larger amounts of data, and true external validation would be performed with data from different studies.

Sources

Hui Zou, Hao Helen Zhang. *On the Adaptive Elastic-Net with a Diverging Number of Parameters. The Annals of Statistics* Vol. 37, No. 4, 1733–1751. DOI:10.1214/08-AOS625. 2009.

Can-Hui Zhang. *Nearly unbiased variable selection under minimax concave penalty. The Annals of Statistics* Vol. 38, No. 2, 894-942. DOI:10.1214/09-AOS729. 2010.

Nan Xiao, Miaozhu Li. *An Introduction to hdnom. R-Project* 2019