Regularized Regression & Corals

S. Eanes

2/21/2020

Data Setup

```
coral <- read.csv("~/School/Fossil Coral/data/coral_3weighted.csv")

#omit data with no response
coral <- coral[!is.na(coral$U238),]

#select three largest species
genus_trim <- c("Acropora", "Porites")
#remove coral with age > 10
coral <- coral[which(coral$Genus %in% genus_trim),]  %>%
  dplyr::filter(Age < 10)
#remove calcite > 1
coral <- coral[(coral$Calcite <= 1 | is.na(coral$Calcite)),]

#clean up a nice dataframe
coral.df <- coral %>% mutate(Temperature = Temp) %>%
  select(U238,pH,TAlk,Salinity,Temperature,OmegaA,TCO2) %>%
  scale() %>%
  data.frame()
nrow(coral.df)
```

```
## [1] 700
```

Bootstrap Lasso Function

We build a function that takes in the number of bootstraps to compute, and the grid of lambda values to input in the glmnet function. This is the penalty paramater that determines the strinkage. The function will cross validate over this range of values to determine the minimum and 1 standard error values. Refer to glmnet documentation for more details. This function outputs a dataframe for each of the bootstrap iterations lambda and MSE values, as well as the coefficients for each iteration.

```
##LASSO
coralboot <- function(n.boot=100,lambda.grid=exp(seq(-10,1,length=200)),verbose=F){

  lambda.min <- numeric(n.boot)
  lambda.1se <- numeric(n.boot)

  mse.min <- numeric(n.boot)
  mse.1se <- numeric(n.boot)

  coef.min <- matrix(nrow=ncol(coral.df),ncol=n.boot)
  rownames(coef.min) <- c("Intercept",colnames(coral.df %>% select(-U238)))
  coef.1se <- matrix(nrow=ncol(coral.df),ncol=n.boot)
  rownames(coef.1se) <- c("Intercept",colnames(coral.df %>% select(-U238)))


  for(i in 1:n.boot){
    if(verbose){print(i)}

    samp <- sample(1:nrow(coral.df),nrow(coral.df),replace = T)
    bootsamp <- coral.df[samp,]
    oob <- coral.df[-samp,]

    cv.lasso <- cv.glmnet(U238 ~ ., data=bootsamp,alpha=1,lambda=lambda.grid,intercept=F)

    lambda.min[i] <- cv.lasso$lambda.min
    lambda.1se[i] <- cv.lasso$lambda.1se

    mod.lasso.min <- glmnet(U238 ~ .,
                            data=bootsamp,alpha=1,lambda=cv.lasso$lambda.min,intercept=F)
    mod.lasso.1se <- glmnet(U238 ~ .,
                            data=bootsamp,alpha=1,lambda=cv.lasso$lambda.1se,intercept=F)


    p1 <- predict(mod.lasso.min,oob)
    p2 <- predict(mod.lasso.1se,oob)

    mse.min[i] <- RMSE(p1,oob$U238)
    mse.1se[i] <- RMSE(p2,oob$U238)


    coef.min[,i] <- coef(mod.lasso.min) %>% as.matrix()
    coef.1se[,i] <- coef(mod.lasso.1se) %>% as.matrix()
  }

  cbind(lambda.min=lambda.min,lambda.1se=lambda.1se,mse.min=mse.min,mse.1se=mse.1se,coef.min=t(coef.min
}
```

Bootstrap

Here, we set a seed to achieve consistent results, then apply the function, parse the data we want from the resulting data frame and construct a figure that summarises the results.

```
set.seed(789)
#adjust n.boot to number of desired bootstrap iterations
cboot <- coralboot(n.boot=200,verbose=F)
cboot$mse.min %>% mean()
```

```
## [1] 0.7506127
```

```r
cboot$mse.1se %>% mean()
```

```
## [1] 0.7845968
```

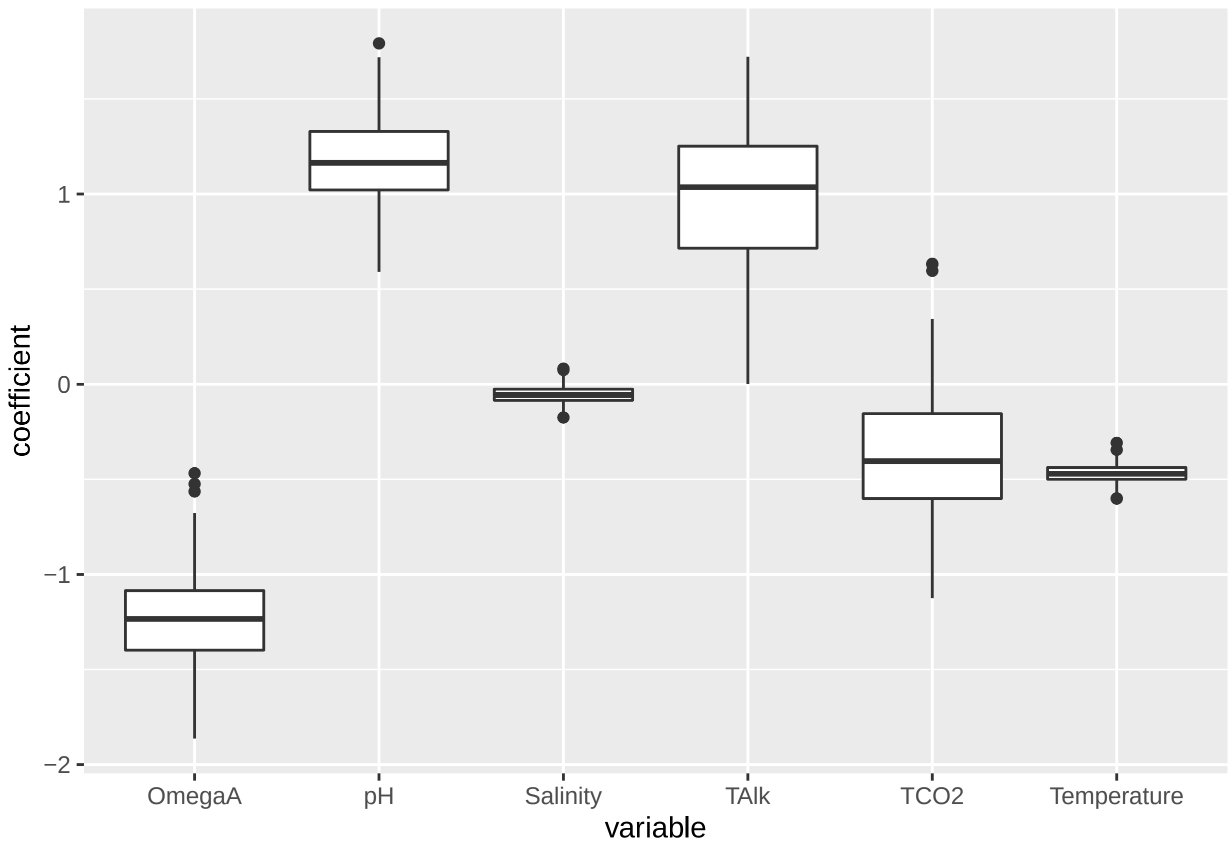
```r
coefs <- cboot %>% select(-lambda.min,-lambda.1se,-mse.min,-mse.1se,-Intercept,-Intercept.1)
means <- colMeans(coefs)
sdevs <- apply(coefs,2,sd)


#the min lambda coefs
coefsMin <- data.frame(coefs[1:(ncol(coral.df)-1)])
#the 1se lambda coefs
coefs1SE <- data.frame(coefs[(ncol(coral.df)):ncol(coefs)])
colnames(coefs1SE) <- colnames(coefsMin)

coefGatherMin <- coefsMin %>% gather() %>% cbind(rep("min",nrow(coefsMin)))
colnames(coefGatherMin) <- c("variable","coefficient","type")
coefGather1SE <- coefs1SE %>% gather() %>% cbind(rep("1se",nrow(coefs1SE)))
colnames(coefGather1SE) <- c("variable","coefficient","type")

coefGatherBoth <- rbind(coefGatherMin,coefGather1SE) %>% data.frame()

#uncomment the following line and the dev.off() line to save an image of the figure locally
#tiff("~/School/Figure8.tiff", width = 10, height = 5, units = 'in', res = 300)
ggplot(coefGatherMin, aes(variable,coefficient))+
  geom_boxplot()
```



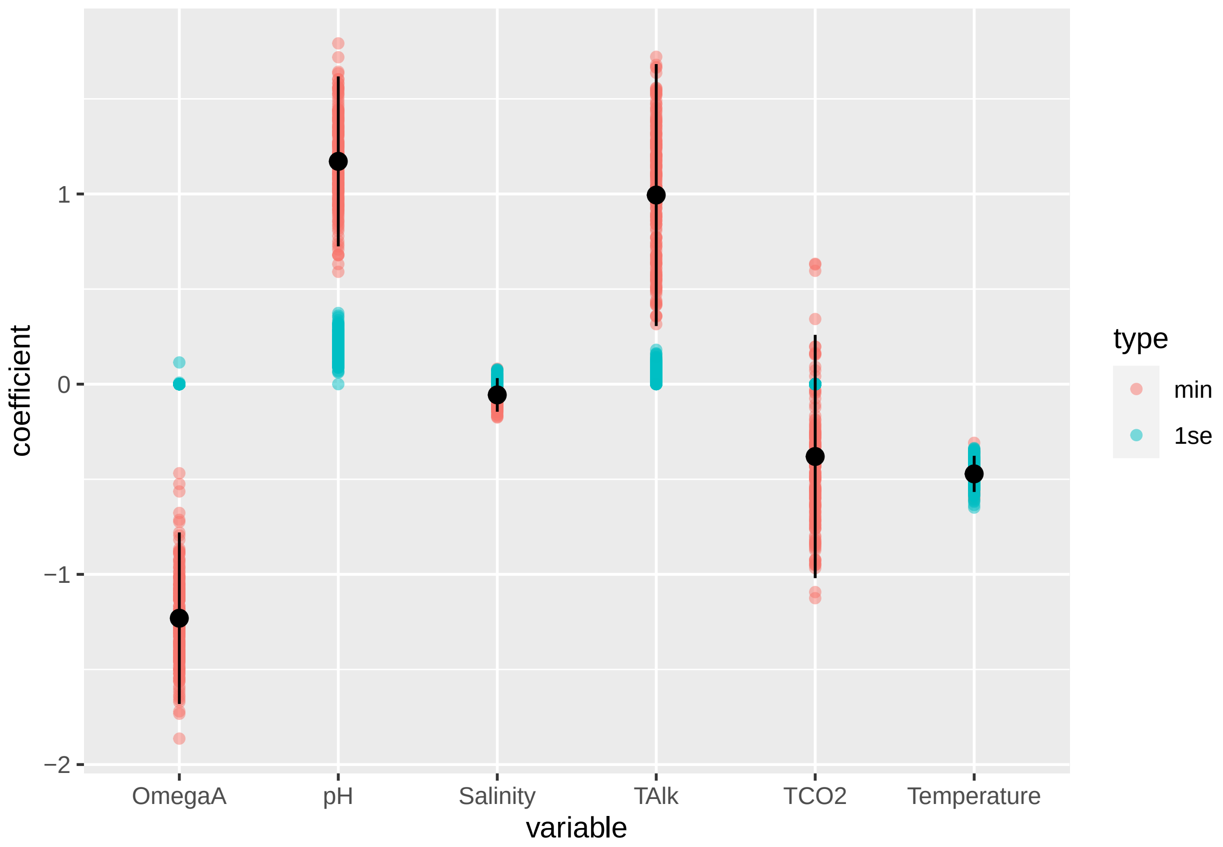
```
#dev.off()
```

Overlay Min and 1SE results

Here we can see that all of the 1SE values tend to cluster near 0, which makes sense because the 1se value of lambda will apply a more aggressive shrinkage.

```
dat1 <- coefGatherBoth %>% filter(type=="min") %>% group_by(variable,type) %>% summarise(n=n(), mean=mea
dat2 <- coefGatherBoth %>% filter(type=="1se") %>% group_by(variable,type) %>% summarise(n=n(), mean=mea

coefGatherBoth %>%
  ggplot(aes(x=variable,color=type))+
  geom_point(aes(y=coefficient),alpha=.5)+
  geom_pointrange(data=dat1,aes(y=mean,ymin=mean-1.96*sd,ymax=mean+1.96*sd),color='black')
```



Best Subset Selection

```
Y <- c("Temperature","Salinity","pH","TCO2","OmegaA","TAlk")

out <- unlist(lapply(1:length(Y), function(n) combn(Y, n, FUN=function(row) paste0("U238 ~ ", paste0(ro
head(out)
```

```
## [1] "U238 ~ Temperature" "U238 ~ Salinity"    "U238 ~ pH"
## [4] "U238 ~ TCO2"        "U238 ~ OmegaA"      "U238 ~ TAlk"
```

```
library(broom)
tmp = dplyr::bind_rows(lapply(out, function(frml) {
  a = glance(lm(frml, data=coral.df))
  a$frml = frml
  return(a)
}))
tmp[order(tmp$AIC),]$frml %>% head()
```

```
## [1] "U238 ~ Temperature+pH+OmegaA+TAlk"
## [2] "U238 ~ Temperature+pH+TCO2+OmegaA+TAlk"
## [3] "U238 ~ Temperature+Salinity+pH+TCO2+OmegaA+TAlk"
## [4] "U238 ~ Temperature+Salinity+pH+OmegaA+TAlk"
## [5] "U238 ~ Temperature+pH+TCO2+OmegaA"
## [6] "U238 ~ Temperature+Salinity+pH+TCO2+OmegaA"
```

```r
tmp[order(tmp$BIC),]$frml %>% head()
```

```
## [1] "U238 ~ Temperature+pH+OmegaA+TAlk"
## [2] "U238 ~ Temperature+pH+TCO2+OmegaA+TAlk"
## [3] "U238 ~ Temperature+Salinity+pH+OmegaA+TAlk"
## [4] "U238 ~ Temperature+pH+TCO2+OmegaA"
## [5] "U238 ~ Temperature+Salinity+pH+TCO2+OmegaA+TAlk"
## [6] "U238 ~ Temperature+Salinity+pH+TCO2+OmegaA"
```