Homework 2

Yuning Li

2024-02-01

library(epifitter)  
library(ggplot2)  
library(dplyr)

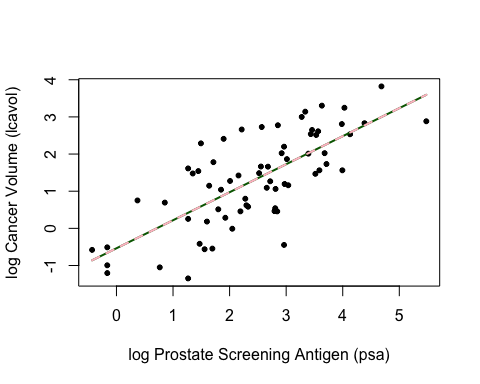
##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

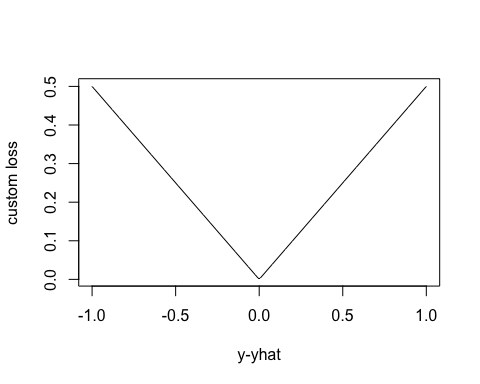
## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(magrittr)  
library(cowplot)

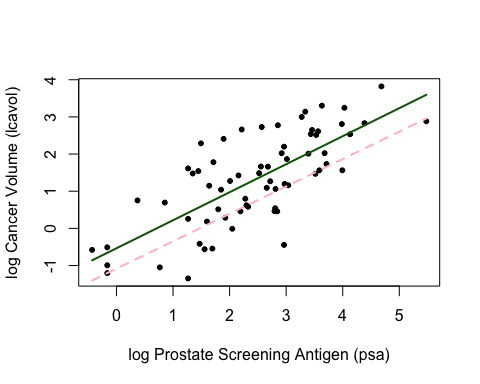
## load prostate data  
prostate <- read.csv("prostate.csv")  
## subset to training examples  
prostate\_train <- subset(prostate, train==TRUE)  
  
## plot lcavol vs lpsa  
plot\_psa\_data <- function(dat=prostate\_train) {  
 plot(dat$lpsa, dat$lcavol,  
 xlab="log Prostate Screening Antigen (psa)",  
 ylab="log Cancer Volume (lcavol)",  
 pch = 20)  
}  
plot\_psa\_data()  
############################  
## regular linear regression  
############################  
  
## L2 loss function  
L2\_loss <- function(y, yhat)  
 (y-yhat)^2  
  
## fit simple linear model using numerical optimization  
## ... - arguments passed to los  
fit\_lin <- function(y, x, loss=L2\_loss, beta\_init = c(-0.51, 0.75), ...) {  
   
 ## function to compute training error  
 err <- function(beta)  
 mean(loss(y, beta[1] + beta[2]\*x, ...))  
   
 ## find value of beta that minimizes training error  
 beta <- optim(par = beta\_init, fn = err)  
   
   
 return(beta)  
}  
  
## make predictions from linear model  
predict\_lin <- function(x, beta)  
 beta[1] + beta[2]\*x  
  
## fit linear model  
lin\_beta <- fit\_lin(y=prostate\_train$lcavol,  
 x=prostate\_train$lpsa,  
 loss=L2\_loss)  
  
## compute predictions for a grid of inputs  
x\_grid <- seq(min(prostate\_train$lpsa),  
 max(prostate\_train$lpsa),  
 length.out=100)  
lin\_pred <- predict\_lin(x=x\_grid, beta=lin\_beta$par)  
  
## plot data  
plot\_psa\_data()  
  
## plot predictions  
lines(x=x\_grid, y=lin\_pred, col='darkgreen', lwd=2)  
  
## do the same thing with 'lm'  
lin\_fit\_lm <- lm(lcavol ~ lpsa, data=prostate\_train)  
  
## make predictins using 'lm' object  
lin\_pred\_lm <- predict(lin\_fit\_lm, data.frame(lpsa=x\_grid))  
  
## plot predictions from 'lm'  
lines(x=x\_grid, y=lin\_pred\_lm, col='pink', lty=2, lwd=2)



##################################  
## try modifying the loss function  
##################################  
  
## tilted absolute loss  
tilted\_abs\_loss <- function(y, yhat, tau) {  
   
 d <- y-yhat  
   
 ifelse(d > 0, d \* tau, d \* (tau - 1))  
}  
  
custom\_loss <- tilted\_abs\_loss  
  
## plot custom loss function  
err\_grd <- seq(-1,1,length.out=200)  
plot(err\_grd, custom\_loss(0, err\_grd, tau=0.50), type='l',  
 xlab='y-yhat', ylab='custom loss')

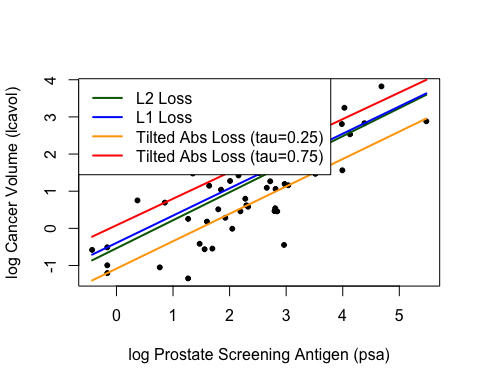


## fit linear model with custom loss  
lin\_beta\_custom <- fit\_lin(y=prostate\_train$lcavol,  
 x=prostate\_train$lpsa,  
 loss=custom\_loss,  
 tau=0.25)  
  
lin\_pred\_custom <- predict\_lin(x=x\_grid, beta=lin\_beta\_custom$par)  
  
## plot data  
plot\_psa\_data()  
  
## plot predictions from L2 loss  
lines(x=x\_grid, y=lin\_pred, col='darkgreen', lwd=2)  
  
## plot predictions from custom loss  
lines(x=x\_grid, y=lin\_pred\_custom, col='pink', lwd=2, lty=2)



#Q1  
## L1 loss function  
L1\_loss <- function(y, yhat)  
 abs(y - yhat)  
  
## Tilted absolute loss function  
tilted\_abs\_loss <- function(y, yhat, tau) {  
 d <- y - yhat  
 ifelse(d > 0, d \* tau, d \* (tau - 1))  
}

#Q2  
## Fit linear model with L1 loss  
lin\_beta\_L1 <- fit\_lin(y=prostate\_train$lcavol,  
 x=prostate\_train$lpsa,  
 loss=L1\_loss)  
  
## Fit linear model with tilted absolute loss (tau = 0.25)  
lin\_beta\_tilted\_025 <- fit\_lin(y=prostate\_train$lcavol,  
 x=prostate\_train$lpsa,  
 loss=tilted\_abs\_loss,  
 tau=0.25)  
  
## Fit linear model with tilted absolute loss (tau = 0.75)  
lin\_beta\_tilted\_075 <- fit\_lin(y=prostate\_train$lcavol,  
 x=prostate\_train$lpsa,  
 loss=tilted\_abs\_loss,  
 tau=0.75)  
  
## Plot data  
plot\_psa\_data()  
  
## Plot predictions for L2 loss  
lines(x=x\_grid, y=lin\_pred, col='darkgreen', lwd=2)  
  
## Plot predictions for L1 loss  
lines(x=x\_grid, y=predict\_lin(x=x\_grid, beta=lin\_beta\_L1$par), col='blue', lwd=2)  
  
## Plot predictions for tilted absolute loss (tau = 0.25)  
lines(x=x\_grid, y=predict\_lin(x=x\_grid, beta=lin\_beta\_tilted\_025$par), col='orange', lwd=2)  
  
## Plot predictions for tilted absolute loss (tau = 0.75)  
lines(x=x\_grid, y=predict\_lin(x=x\_grid, beta=lin\_beta\_tilted\_075$par), col='red', lwd=2)  
  
## Add legend  
legend("topleft", legend=c("L2 Loss", "L1 Loss", "Tilted Abs Loss (tau=0.25)", "Tilted Abs Loss (tau=0.75)"), col=c('darkgreen', 'blue', 'orange', 'red'), lty=1, lwd=2)



#Q3  
## Fit nonlinear model  
fit\_nonlinear <- function(y, x, loss=L2\_loss, beta\_init = c(-1.0, 0.0, -0.3), ...) {  
 err <- function(beta)  
 mean(loss(y, beta[1] + beta[2] \* exp(-beta[3] \* x), ...))  
 beta <- optim(par = beta\_init, fn = err)  
 return(beta)  
}  
  
## Make predictions from nonlinear model  
predict\_nonlinear <- function(x, beta)  
 beta[1] + beta[2] \* exp(-beta[3] \* x)

#Q4  
## Fit nonlinear model with L2 loss  
nonlin\_beta\_L2 <- fit\_nonlinear(y=prostate\_train$lcavol,  
 x=prostate\_train$lpsa,  
 loss=L2\_loss)  
  
## Fit nonlinear model with L1 loss  
nonlin\_beta\_L1 <- fit\_nonlinear(y=prostate\_train$lcavol,  
 x=prostate\_train$lpsa,  
 loss=L1\_loss)  
  
## Fit nonlinear model with tilted absolute loss (tau = 0.25)  
nonlin\_beta\_tilted\_025 <- fit\_nonlinear(y=prostate\_train$lcavol,  
 x=prostate\_train$lpsa,  
 loss=tilted\_abs\_loss,  
 tau=0.25)  
  
## Fit nonlinear model with tilted absolute loss (tau = 0.75)  
nonlin\_beta\_tilted\_075 <- fit\_nonlinear(y=prostate\_train$lcavol,  
 x=prostate\_train$lpsa,  
 loss=tilted\_abs\_loss,  
 tau=0.75)  
  
## Plot data  
plot\_psa\_data()  
  
## Plot predictions for nonlinear model with L2 loss  
lines(x=x\_grid, y=predict\_nonlinear(x=x\_grid, beta=nonlin\_beta\_L2$par), col='darkgreen', lwd=2)  
  
## Plot predictions for nonlinear model with L1 loss  
lines(x=x\_grid, y=predict\_nonlinear(x=x\_grid, beta=nonlin\_beta\_L1$par), col='blue', lwd=2)  
  
## Plot predictions for nonlinear model with tilted absolute loss (tau = 0.25)  
lines(x=x\_grid, y=predict\_nonlinear(x=x\_grid, beta=nonlin\_beta\_tilted\_025$par), col='orange', lwd=2)  
  
## Plot predictions for nonlinear model with tilted absolute loss (tau = 0.75)  
lines(x=x\_grid, y=predict\_nonlinear(x=x\_grid, beta=nonlin\_beta\_tilted\_075$par), col='red', lwd=2)  
  
## Add legend  
legend("topleft", legend=c("Nonlinear (L2 Loss)", "Nonlinear (L1 Loss)", "Nonlinear Tilted Abs (tau=0.25)", "Nonlinear Tilted Abs (tau=0.75)"), col=c('darkgreen', 'blue', 'orange', 'red'), lty=1, lwd=2)

