Homework 2

Yuning Li

2024-02-01

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

```
##
## Attaching package: 'dplyr'
```

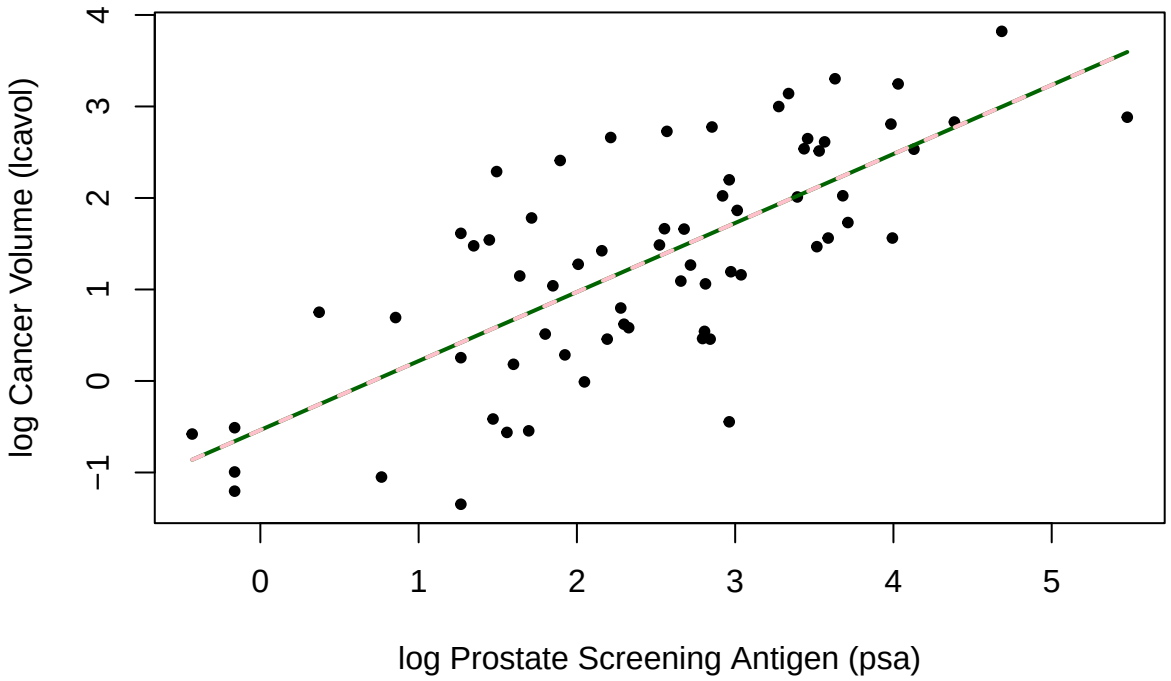
```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

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

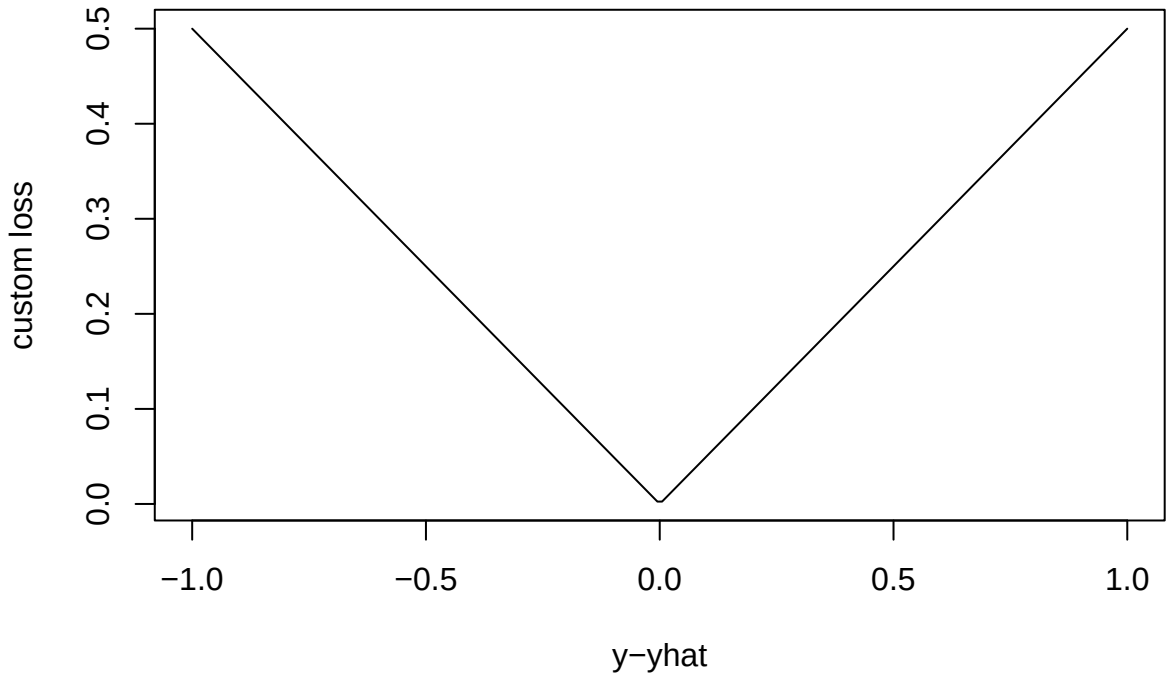
```
library(magrittr)
library(cowplot)
```

```
## 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
```

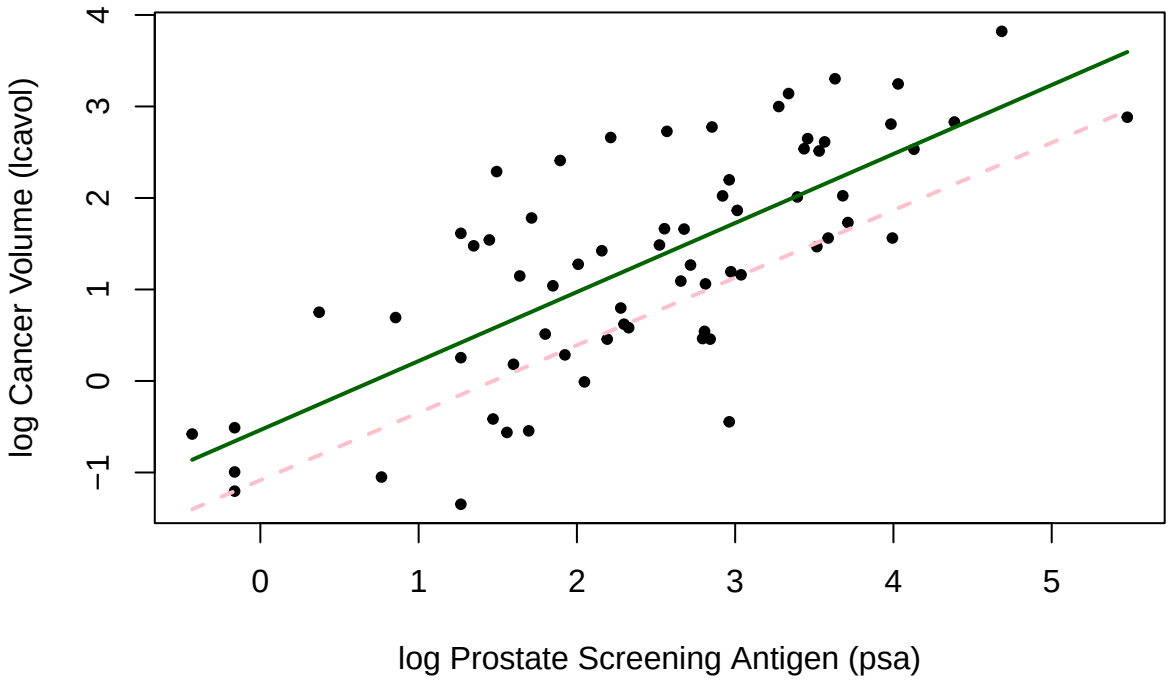
```
  err <- function(beta)
    mean(loss(y,  beta[1] + beta[2]*x, ...))

  ## 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')
```



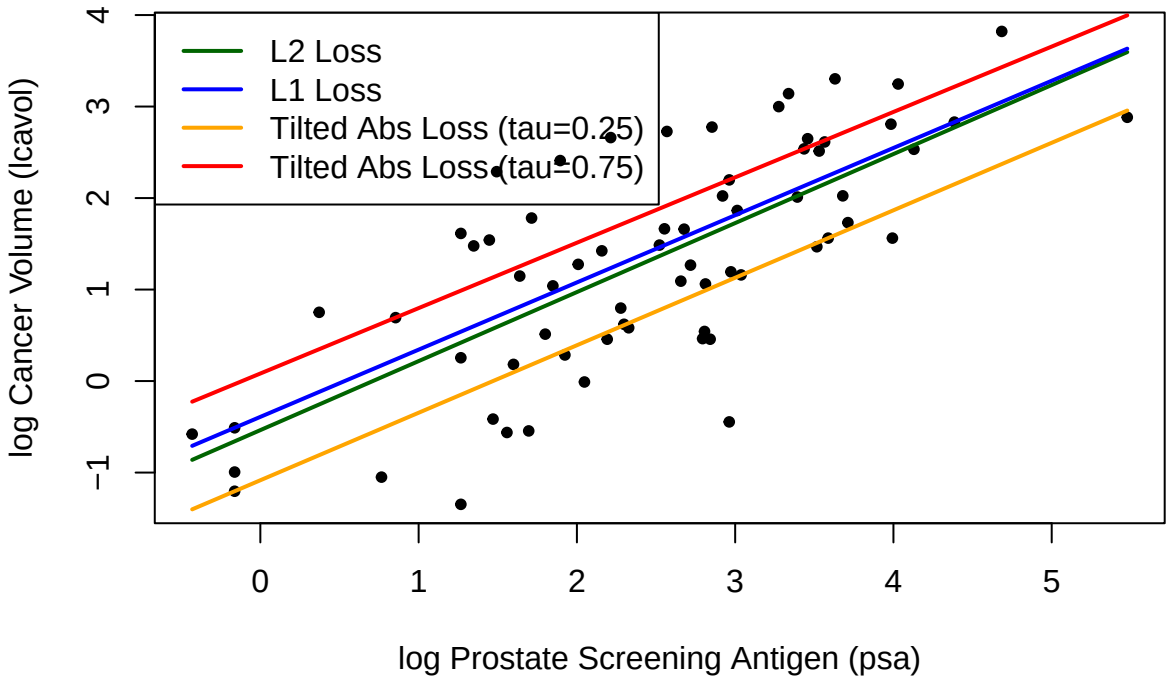

```
## fit linear model with custom loss
lin_beta_custom <- fit_lin(y=prostate_train$lcavol,
                           x=prostate_train$lpsa,
                           loss=custom_loss,
                           tau=0.25)

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
```

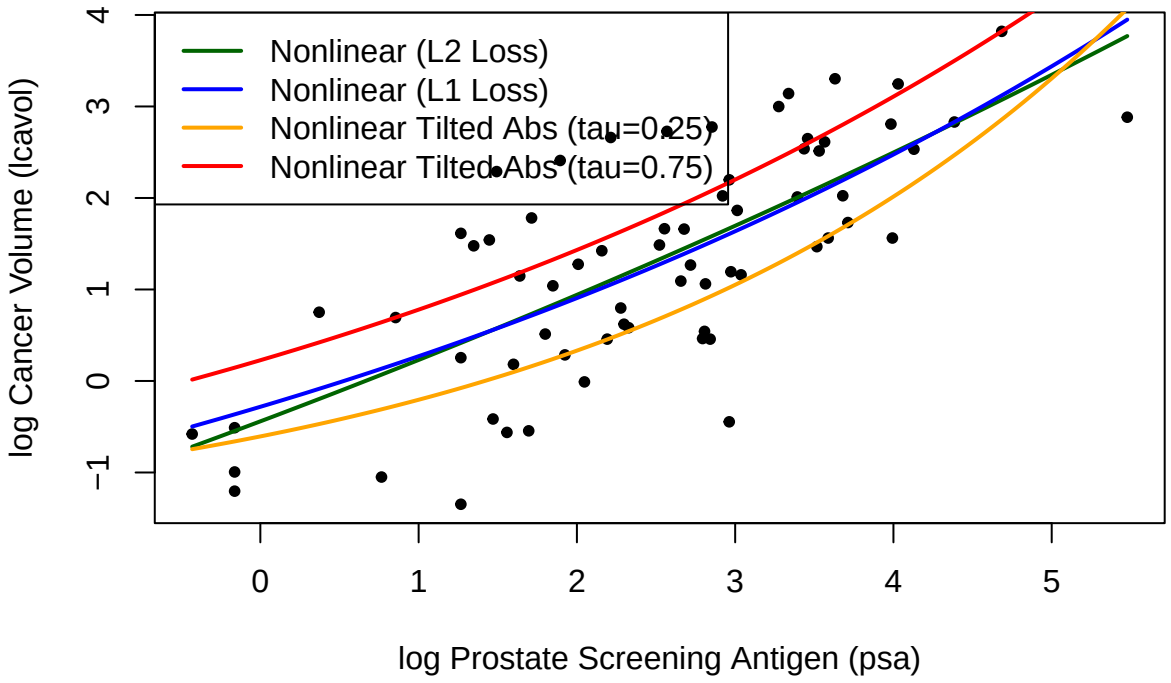
```r
lines(x=x_grid, y=lin_pred, col='darkgreen', lwd=2)

## 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.
```




```r
#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)
```

```r
#Q4
## Fit nonlinear model with L2 loss
nonlin_beta_L2 <- fit_nonlinear(y=prostate_train$lcavol,
                                x=prostate_train$lpsa,
                                loss=L2_loss)
```

```r
## 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 (L2 Loss)
Nonlinear (L1 Loss)
Nonlinear Tilted Abs (tau=0.25)
Nonlinear Tilted Abs (tau=0.75)
log Cancer Volume (lcavol)
log Prostate Screening Antigen (psa)
−1
0
1
2
3
4
0
1
2
3
4
5