STAT/MATH 495: Problem Set 03

Sara Culhane

2017-09-26

Contents

Question

For both `data1` and `data2` tibbles (a tibble is a data frame with some metadata attached):

- Find the splines model with the best out-of-sample predictive ability.
- Create a visualizaztion arguing why you chose this particular model.
- Create a visualizaztion of this model plotted over the given (x_i, y_i) points for $i = 1, \ldots, n = 3000$.
- Give your estimate $\widehat{\sigma}$ of σ where the noise component ϵ_i is distributed with mean 0 and standard deviation σ.

Cross Validation function for k=5

```
set.seed(40) # Set random seed for reproducibility

# Initialize the RMSE vector
r <- rep(0,5)
cv5 <- function(data,df) { # data as full dataset and df as degrees of freedom for spline
  for (i in 1:5) {
    # These indices indicate the interval of the test set
    # Algorithm takes first slice 1 to 600 then 601 to 1200, 1200 to 1800, 1800 to 2400 and 2400
    # The "slice" becomes the test set for that fold of the CV and we remove the same slice from
    slice<- (((i-1) * round((1/5)*nrow(data))) + 1):((i*round((1/5) * nrow(data))))


    train <- data[-slice,] #Remove slice from train


    test <- data[slice,] # Keep only corresponding rows from slice for test

    # Fit spline model using each training set

    mod <- smooth.spline(x=train$x, y=train$y, df=df)
    output <- predict(mod, test$x) %>%
      as_data_frame()
```

```
    r[i]<- sqrt(mean((test$y-output$y )^2))

  }
return(sum(r)/length(r)) # Return mean RMSE for all 5 folds
}
```

Calculation of MSE for any df

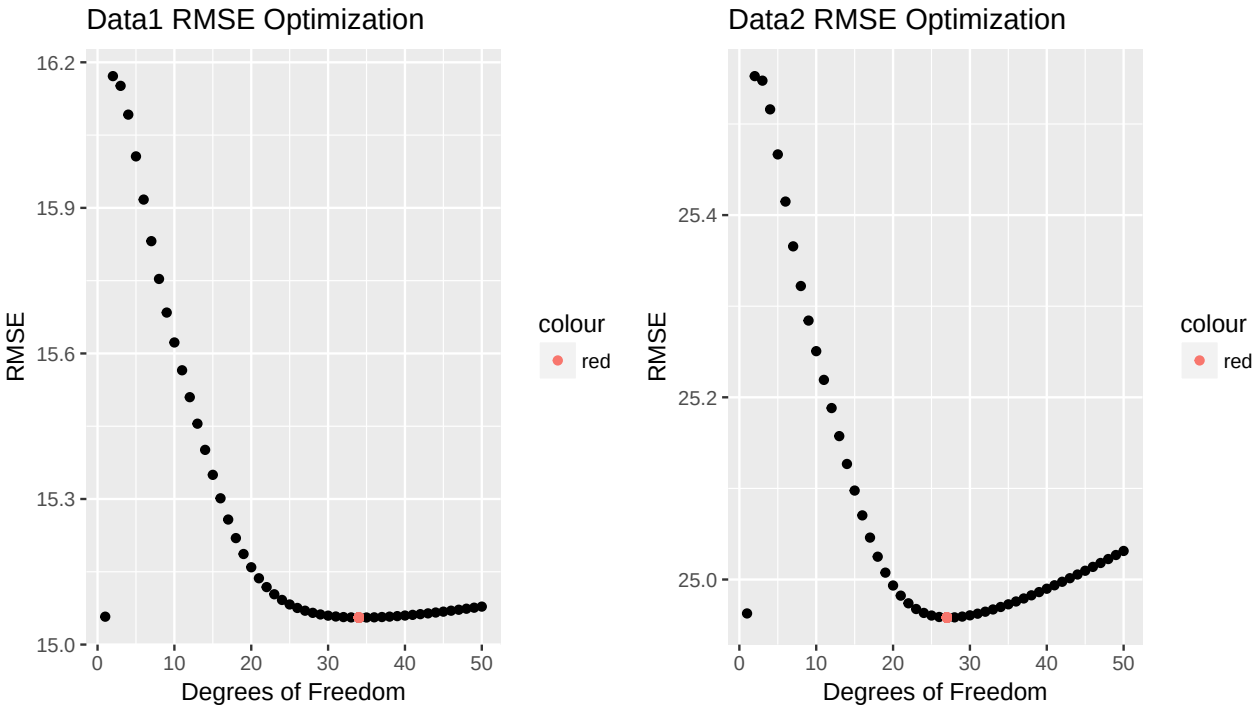
```
error <- c()
# This is a function to create a vector of RMSE for each df that we want to look at

m1 <- function(data,n) {
  for (i in 1:n) {
   error[i] <- cv5(data,i) # stores the mean RMSE from each df
  }
  return(error)
}
```

Finding optimal df for both Data1 and Data2

```
# Calculate RMSE for df 1 to 50

d1 <- m1(data1,50)
d2 <- m1(data2,50)

# Create a data frame for plotting purposes

dp1 <- data.frame(x =1:50, y=d1)
dp2 <- data.frame(x =1:50, y=d2)

# Use which.min to find the df with lowest RMSE

low1 <- which.min(m1(data1,50))
low2 <- which.min(m1(data2,50))
```

Plots of RMSE for all df examined

```
p1 <- ggplot(dp1, aes(x=x)) +
  geom_point(aes(y=y)) + xlab("Degrees of Freedom") + ylab("RMSE") +
  geom_point(aes(x=low1,y=15.05571, color="red")) + labs(title="Data1 RMSE Optimization")
p2 <-  ggplot(dp2, aes(x=x)) +
  geom_point(aes(y=y))+ xlab("Degrees of Freedom") + ylab("RMSE") + labs(title="Data2 RMSE Optimization
  geom_point(aes(x=low2,y=24.95825, color="red"))
grid.arrange(p1, p2,ncol=2)
```



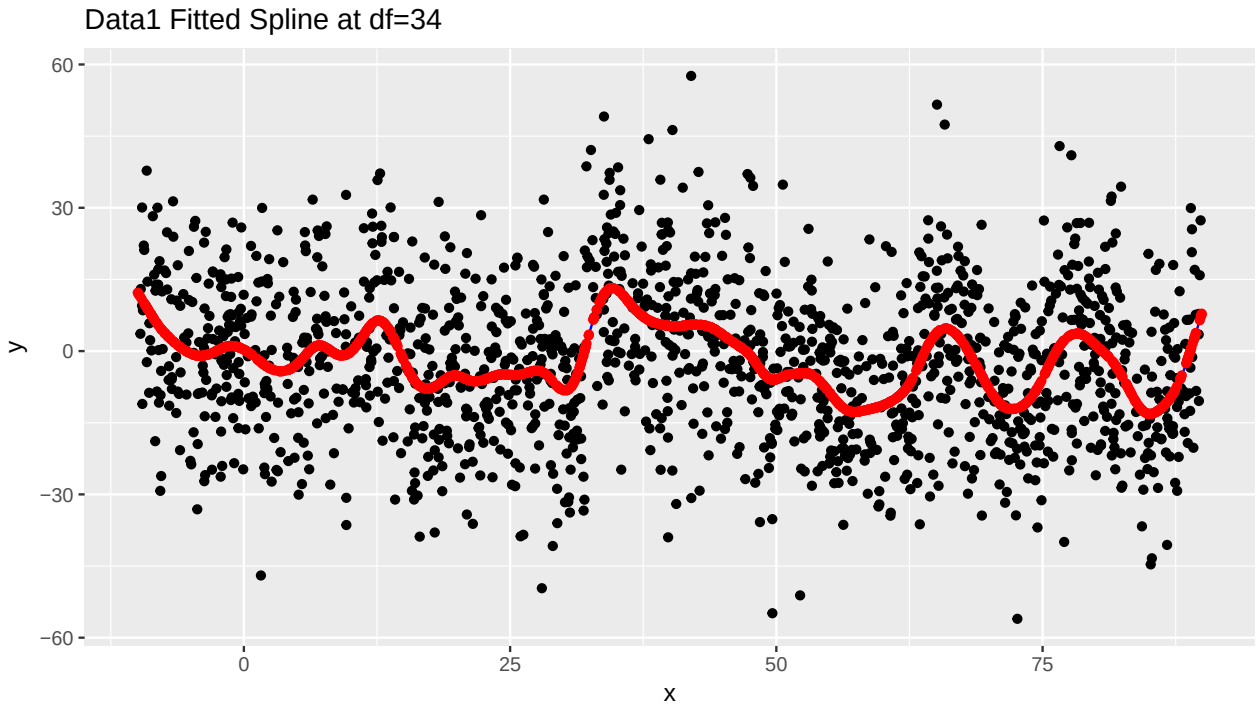
Based on analyis of degrees of freedom from 1 to 50 using our CV 5-fold calculations, we will use 34 and 27 df for data1 and data2 respectively for our df in our spline fit. These points are colored red.

Application to full dataset

With this plot, we can make estimates for $f(x)$, which is shown by our fitted spline.

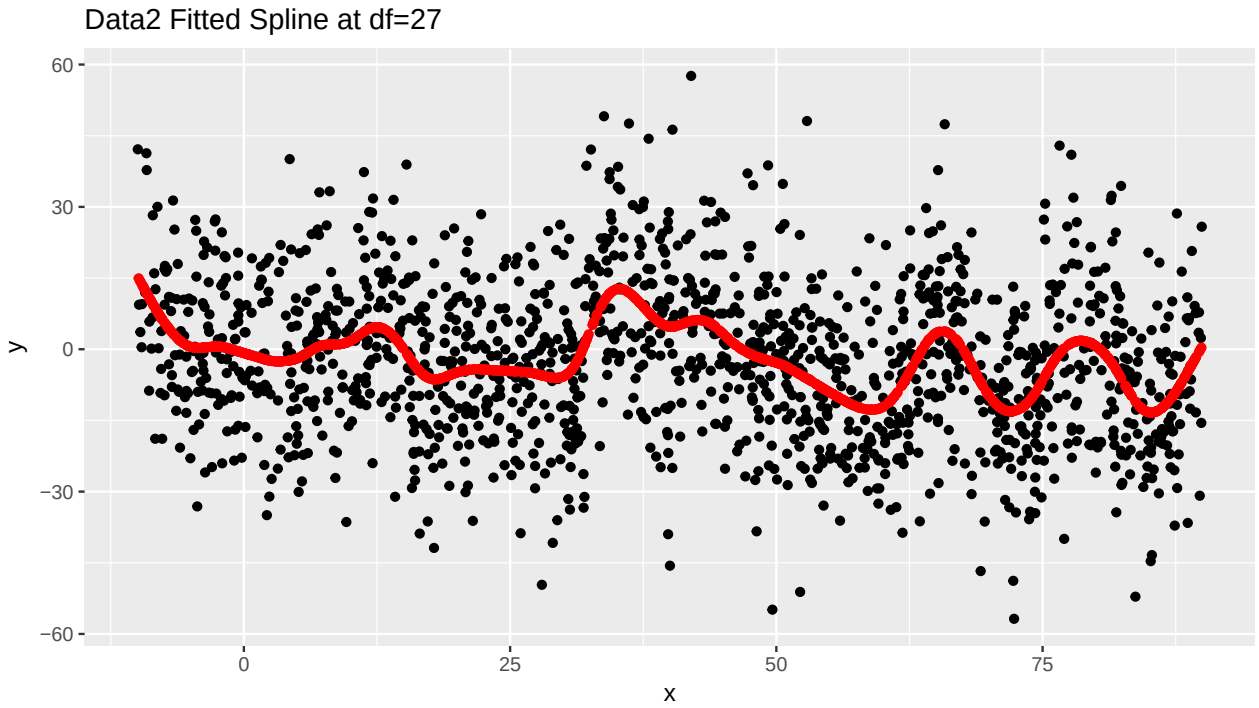
Data1 based on df=34

```
train1 <- data1 %>% sample_n(nrow(data1)/2)
test1 <- anti_join(data1,train1,by="ID")
mod1 <- smooth.spline(x=train1$x, y=train1$y, df=low1)
output1 <- predict(mod1, test1$x) %>%
  as_data_frame()

mod_t1 <- mod1 %>%
  broom::augment()
p3 <- ggplot(mod_t1, aes(x=x)) +
  geom_point(aes(y=y)) +
  geom_line(aes(y=.fitted), col="blue")
p3 +
  geom_point(data=output1, aes(x=x, y=y), col="red") +
  labs(title="Data1 Fitted Spline at df=34")
```



Data2 based on df=27

```
train2 <- data1 %>% sample_n(nrow(data2)/2)
test2 <- anti_join(data2,train2,by="ID")
mod2 <- smooth.spline(x=train2$x, y=train2$y, df=low2)
output2 <- predict(mod2, test2$x) %>%
  as_data_frame()

mod_t2 <- mod2 %>%
  broom::augment()
p4 <- ggplot(mod_t2, aes(x=x)) +
  geom_point(aes(y=y)) +
  geom_line(aes(y=.fitted), col="blue")
p4 +
  geom_point(data=output2, aes(x=x, y=y), col="red") +
  labs(title="Data2 Fitted Spline at df=27")
```



Estimating error from σ

```
sigma1 <- sd(test1$y - output1$y)
sigma2 <- sd(test2$y - output2$y )
set.seed(40)
n <- 3000
ep1 <- rnorm(n,0,sigma1)
ep2 <- rnorm(n,0,sigma2)
meanep1 <- mean(ep1)
meanep2 <- mean(ep2)
```

Thus, we have that the ϵ have means 0.3437428 and -0.3042762 for Data1 and Data2 respectively.