STATS - HW 3

Wenjing Li

2/20/2020

Question 1

1(1)

g1 will be smaller. Since the second term means the smoothness and the first term means the fitness. g_2 has better smoothness compared with g_1. g_2 will do a better job at testing and g_1 wil be better at training.

1(2)

g2 will be smaller. The reason is above.

1(3)

If $\lambda = 0$, then $g_1 = g_2$. And g_1, g_2 are equal to the MSE function. The training error will be smaller, and the testing error will be larger than above g that has been selected. ## Question 2

2(1)

At first, we need to take a look at our data set.

```
# Make a summary for the current data set.
summary(dat)
```

```
##      ozone         radiation      temperature         wind
##  Min.   :  1.0   Min.   :  7.0   Min.   :57.00   Min.   : 2.300
##  1st Qu.: 18.0   1st Qu.:113.5   1st Qu.:71.00   1st Qu.: 7.400
##  Median : 31.0   Median :207.0   Median :79.00   Median : 9.700
##  Mean   : 42.1   Mean   :184.8   Mean   :77.79   Mean   : 9.939
##  3rd Qu.: 62.0   3rd Qu.:255.5   3rd Qu.:84.50   3rd Qu.:11.500
##  Max.   :168.0   Max.   :334.0   Max.   :97.00   Max.   :20.700
```

From the above result, all the data has been cleaned and we can fit the model directly.

Next, split the data set into training data and testing data and fit a linear model.

```
# Split data - 70% as training data & 30% as testing data
set.seed(123)
n = dim(dat)[1]
train_id = sample(seq(1, n, 1), floor(n*0.7))
test = dat[-train_id, ]
train = dat[train_id, ]

# Fit linear model based on training data
lm = lm((ozone)^(1/3) ~ ., data = train)
summary(lm)
```

```
##
## Call:
## lm(formula = (ozone)^(1/3) ~ ., data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.94503 -0.40230 -0.00071  0.27566  1.50475
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.8521380  0.6449384  -1.321 0.190538
## radiation    0.0016477  0.0006398   2.575 0.012037 *
## temperature  0.0579790  0.0071750   8.081 9.93e-12 ***
## wind        -0.0656469  0.0180658  -3.634 0.000516 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4999 on 73 degrees of freedom
## Multiple R-squared:  0.7111, Adjusted R-squared:  0.6992
## F-statistic:  59.9 on 3 and 73 DF,  p-value: < 2.2e-16
```

From the linear model result, three predictors are all significant. The final linear model based on the training data is:

$ozone^{\frac{1}{3}} = -0.852 + 0.002 * radiation + 0.058 * temperature - 0.066 * wind$

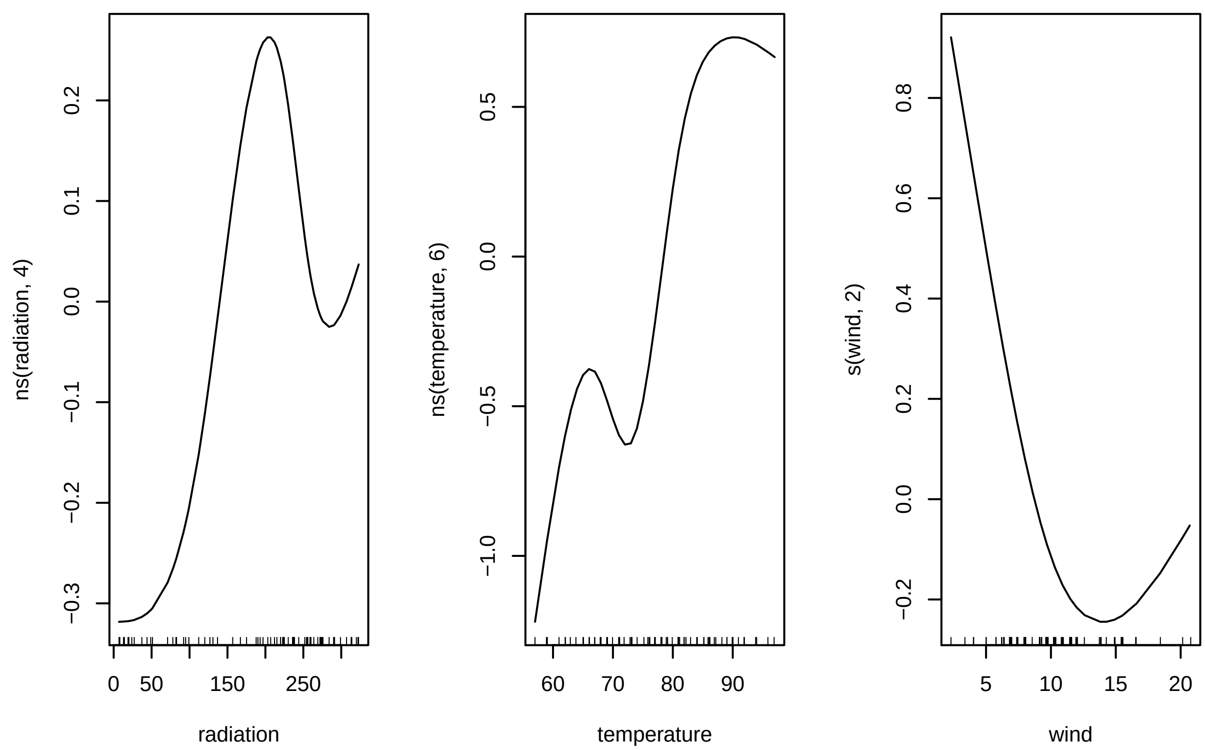
The p-value of the model is less than 2.2e-16, which means the model is significant. The R-squared is 0.71 which is relatively large. Overall, the linear model gives us a good fit.

2(2)

Next, I fitted a GAM on the training data and got results as below.

```r
# Fit GAM model
Kfold_CV <- function(K,train,h,j,k) {
  fold_size = floor(nrow(train)/K)
  cv_error = rep(0,K)
  for(i in 1:K) {
    if(i != K) {
      CV_test_id = ((i-1)*fold_size+1):(i*fold_size)
    }else{
      CV_test_id = ((i-1)*fold_size+1):nrow(train)
    }
    CV_train = train[-CV_test_id,]
    CV_test = train[CV_test_id,]
    # Fit gam
    gam_CV = gam((ozone)^(1/3) ~ ns(radiation,h)+ns(temperature,j)+
                   ns(wind,k), data=CV_train)
    pred_CV = predict.Gam(gam_CV, CV_test)
    # Calculate CV error by taking averages
    cv_error[i] = mean((CV_test$ozone - pred_CV)^2)
  }
  return(mean(cv_error))
}
```

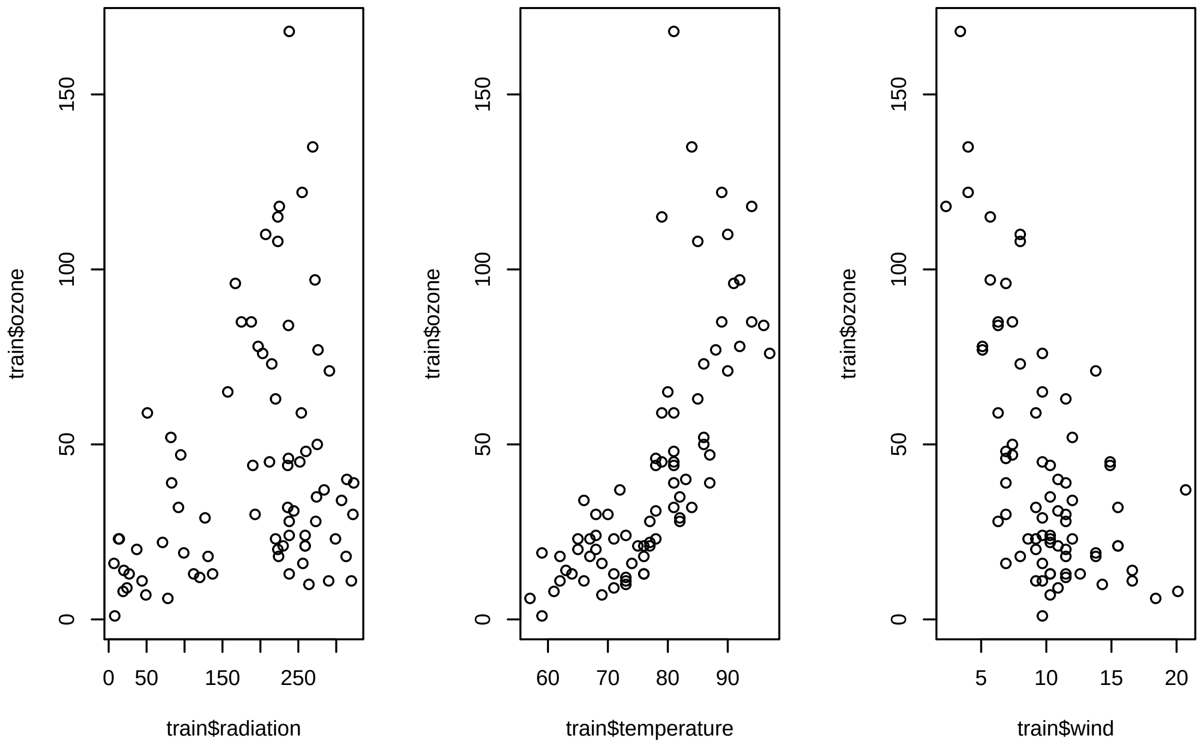
```
K = 10
mse = rep(0, 2000)
dim(mse) = c(10,10,20)
fold_size = floor(nrow(train)/K)
cv_error = rep(0,K)
for (h in 1:10) {
 for (j in 1:10) {
   for (k in 1:20) {
     mse[h,j,k] = Kfold_CV(K,train,h,j,k)
   }
 }
}

min_error = min(mse)

gam = gam((ozone)^(1/3) ~ ns(radiation,4)+ns(temperature,6)+s(wind,2), data=train )
par(mfrow = c(1,3))
plot.Gam(gam)
```



That seems a good fit to the raw data.

```
par(mfrow = c(1,3))
plot(train$radiation, train$ozone)
plot(train$temperature, train$ozone)
```

```r
plot(train$wind, train$ozone)
```



2(c)

```r
#compute mse_gam
pred_gam = predict.Gam(gam, test)
mse_gam = mean((pred_gam - test$ozone)^2)

#compute mse_lm
pred_lm = predict.lm(lm,test)
mse_lm = mean((pred_lm - test$ozone)^2)

#create table
table_error = c(mse_lm, mse_gam)
dim(table_error) = c(1,2)
colnames(table_error) = c("Linear Model", "GAM")
rownames(table_error) = c("MSE")
cap = paste("*Testing and Training Error for LM and GAM*")
knitr::kable(table_error, caption = cap)
```

Table 1: *Testing and Training Error for LM and GAM*

	Linear Model	GAM
MSE	2012.594	2003.539

The MSE of the linear model is 2012.6, is larger than the MSE of the GAM model.

2(d)

From the raw data plot, the predictor "radiation" is not linear with the response "ozone". The other two predictors can be seen as linear or non-linear predictors. And From the MSE above, the GAM model provides a better fit and choosing the non-linear relationship is better than the linear relationship.