HW7 q3

10/30/2019

a)

Suppose we use an additive model where each additive function is a univariate kernel smoother. Do you expect this model would have more or less bias and variance than the full kernel smoother? Be specific about both the bias and variance.

The full kernel smoother will not assume any relationships between variables, whereas in teh additive model each univariate kernel smoother only estimates the effect of one term. The additive model should have higher bias than the full kernel smoother since assumptions are made about how the terms are related. However the additive model will have lower variance since it is less flexible to noise than the full kernel smoother.

b)

Split the data into training and test sets, as you did in Homework 6. You should not need to subsample the data, as additive models are quite fast. Use predict and compare the squared-error loss of the additive model to that of the linear model. Does the additive model do dramatically better?

The mse of the additive model is 2255.892 which is better than 2435.305 for the linear model. However this is not a dramatic difference. The only variables we actually smoothed were DEP_TIME, DAY_OF_MONTH and DAY_OF_WEEK. This suggests the data can be fit relatively well using just a linear model involving these variables as well as the qualitative variables.

```
# Run the same baseline linear regression you used in Homework 6, so we have a basis to
# compare against.
set.seed(1234)
base=lm(ARR_DELAY~ DEP_TIME +as.factor(ORIGIN) +
              DAY_OF_MONTH +as.factor(CARRIER) +
              DAY_OF_WEEK, data=train) #[rows,]
#summary(base)

#Using the gam function in the mgcv package, fit an additive model to the training data using the defau
additive <- mgcv::gam(ARR_DELAY~s(DEP_TIME) +as.factor(ORIGIN) +
              s(DAY_OF_MONTH) +as.factor(CARRIER) +
              s(DAY_OF_WEEK, k=7),
                      #family=Gamma(link=),
                      data =train)

# (You should smooth over day of week and day of month. But note that you may need to
# set the k argument to s() for day of week; by default, mgcv tries to use a spline basis with more bas

# Use predict and compare the squared-error loss of the additive model to that of the linear model
mean((test$ARR_DELAY - predict.lm(base, newdata = test))^2, na.rm = TRUE) #2435.305
```

[1] 2772.192

```
mean((test$ARR_DELAY - predict.gam(additive, newdata = test))^2, na.rm = TRUE) # 2133.988
```

[1] 2465.022

c)

The plot method for GAM fits from mgcv plots the smoothed features automatically, including standard errors. Make the plots and interpret the results. What do the plots suggest about the appropriateness of linear regression? Do the plots explain the difference in performance between lienar regression and the additive model? Be sure to interpret the standard error ranges and say what their widths imply.

The plots show variance in the smoothed term (in y axis) versus each of the quantitative variables (on x axis). The plots suggest that:

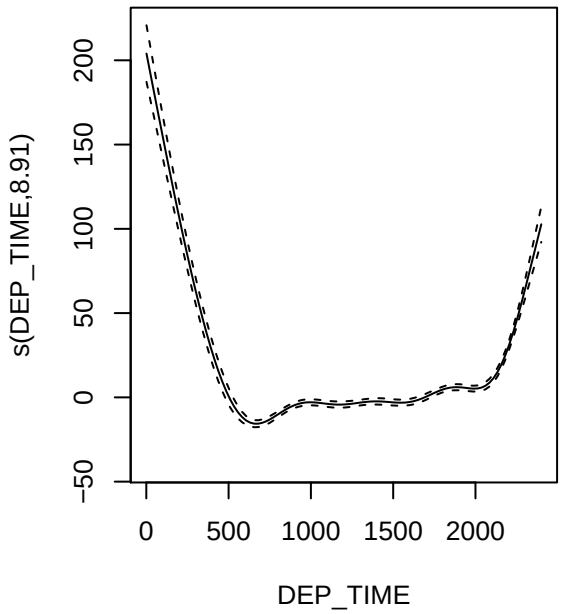
- the relationship between departure time and the arrival delay is nonlinear especially before 5:00 or above 20:00 hours.
- the arrival delay has a somewhat sinusoidal relationship with the day of month.
- arrival delay peaks around every fourth day of the month. This might not be so meaningful since the DAY_OF_WEEK variable does not correspond to weekday but rather how many days it has been since the first of the month.

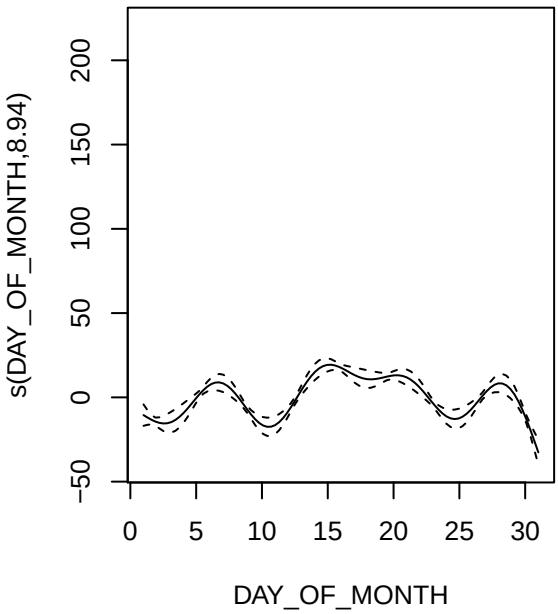
The error ranges are larger for day of month and day of week than for departure time. The small error ranges throughout the range of DEP_TIME suggest that we can be relatively confident in this nonlinear relationship. However, we know the relationship with day of week and day of month less precisely since there is more variation in the response for these predictors in the data we have.

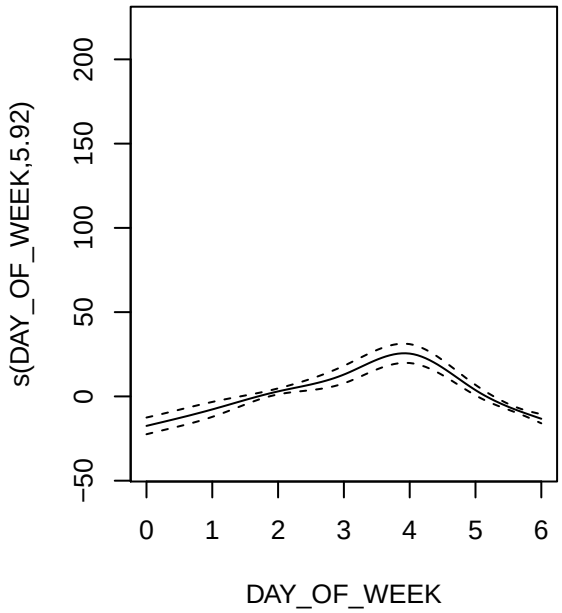
It seems the relationship of arrival delay with departure time has significant nonlinearities, but other than that simple linear regression may be about sufficient. This may explain why the additive model performs better than linear regression, but not by much.

```
# base plot
# par(mfrow=c(3,2))
# plot(summary(base), data =data)

# additive plot
par(mfrow=c(2,2))
plot.gam(additive)
```


s(DEP_TIME,8.91)
−50
0
50
100
150
200
0
500
1000
1500
2000
DEP_TIME


s(DAY_OF_MONTH,8.94)
−50
0
50
100
150
200
0
5
10
15
20
25
30
DAY_OF_MONTH


s(DAY_OF_WEEK,5.92)
−50
0
50
100
150
200
0
1
2
3
4
5
6
DAY_OF_WEEK