homework3

Qifan Wang

2/21/2020

1. When lambda goes to infinity, g1 won't penalize any quadratic terms and g2 goes even further, putting no penalty on cubic terms. g2 will definitely give a more complex model.

Therefore, in this scenario g1 will result in a global quadratic function and g2 will give something very similar to a cubic spline.

For sure, the model that g2 ends up with is more complicated and the training error of g2 is lower.

2. It depends on what the data looks like. Both are possible.
3. When lambda equals to 0, both will result in a interpolation curve and errors will not differ.

```r
ozone=read.csv("ozone_data.txt",sep="\t")
```

```r
summary(ozone)
```

```
##      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
```

```r
set.seed(1)
ozone_data=ozone
ozone_data$ozone=(ozone$ozone)^(1/3)
train_id = sample(1:nrow(ozone_data),floor(0.7*nrow(ozone_data)))
train = ozone_data[train_id,]
test = ozone_data[-train_id,]
```

```r
# cubic root  regression
lm.ozone=lm(ozone~radiation+temperature+wind, data=train)
summary(lm.ozone)
```

```
##
## Call:
## lm(formula = ozone ~ radiation + temperature + wind, data = train)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -1.21744 -0.37482 -0.07101  0.34530  1.34042
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.8894090  0.7558399   1.177  0.24313
## radiation    0.0021651  0.0007692   2.815  0.00627 **
## temperature  0.0383507  0.0084449   4.541 2.16e-05 ***
## wind        -0.0981429  0.0196878  -4.985 4.04e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5268 on 73 degrees of freedom
## Multiple R-squared:  0.6612, Adjusted R-squared:  0.6473
## F-statistic: 47.49 on 3 and 73 DF,  p-value: < 2.2e-16
```

The regression results in a R-squared of 0.6612, and with all the parameters significant. Generally speaking, the model fits pretty nice.

```r
# Fit gam
library(gam)
```

```
## Loading required package: splines
```

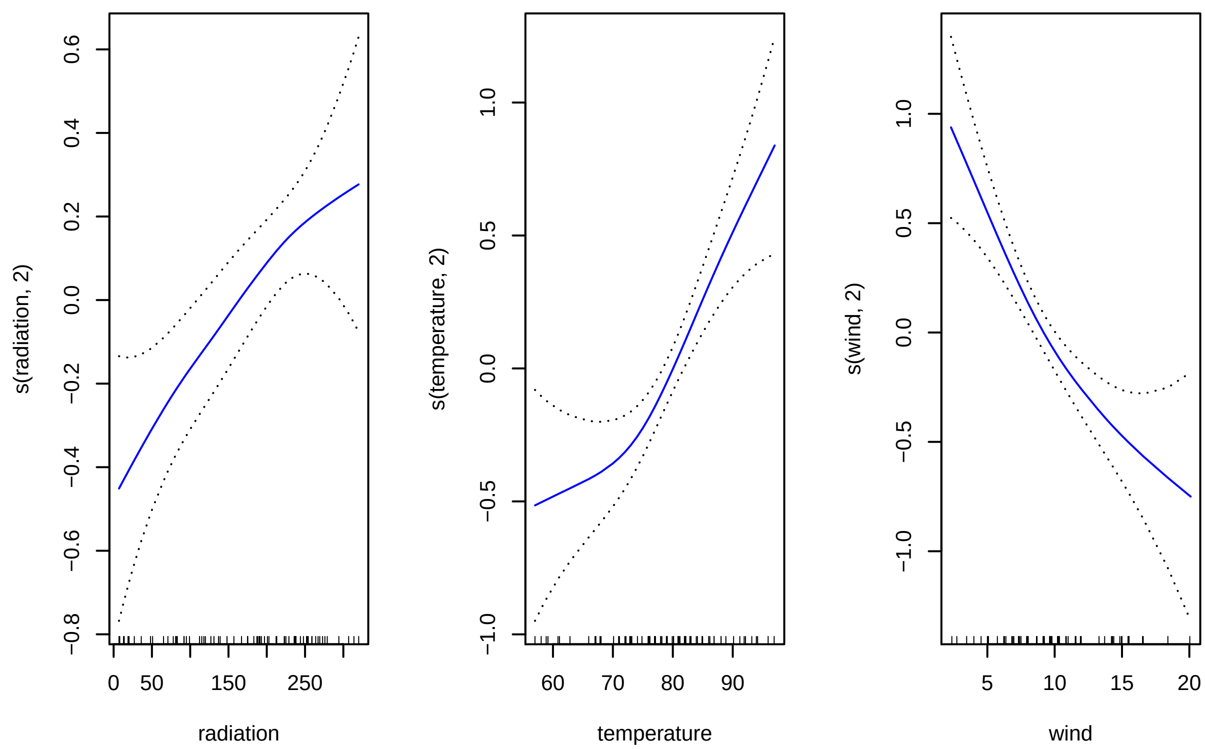
```
## Loading required package: foreach
```

```
## Loaded gam 1.16.1
```

```r
gam.ozone2 = gam(ozone~s(radiation,2)+s(temperature,2)+s(wind,2), data=train)
```

```
## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts argument
## ignored
```

```r
par(mfrow=c(1,3))
plot.Gam(gam.ozone2 ,col="blue",se=T)
```




```r
summary(gam.ozone2)
```

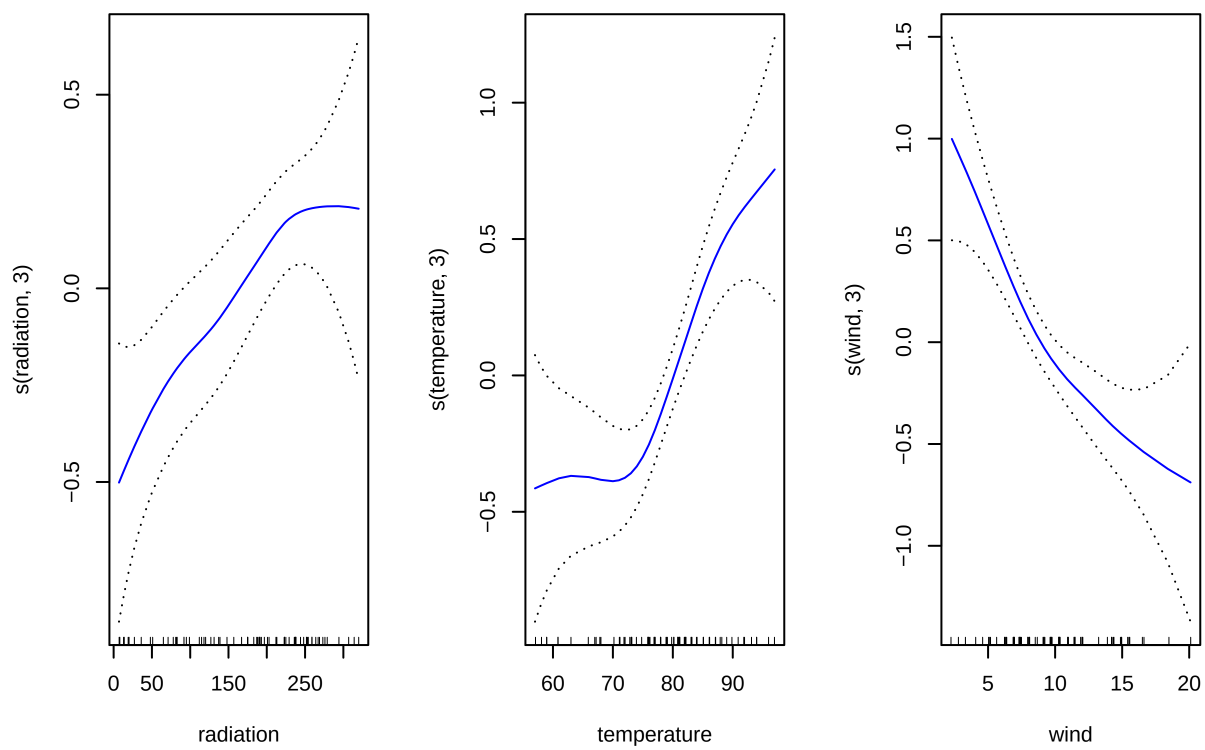
```
##
## Call: gam(formula = ozone ~ s(radiation, 2) + s(temperature, 2) + s(wind,
##     2), data = train)
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.31494 -0.27558 -0.07477  0.34400  1.21110
##
## (Dispersion Parameter for gaussian family taken to be 0.2443)
##
##     Null Deviance: 59.8074 on 76 degrees of freedom
## Residual Deviance: 17.1 on 70 degrees of freedom
## AIC: 118.6528
##
## Number of Local Scoring Iterations: 2
##
## Anova for Parametric Effects
##                   Df  Sum Sq Mean Sq F value    Pr(>F)
## s(radiation, 2)    1 13.6285 13.6285  55.789 1.730e-10 ***
## s(temperature, 2)  1 18.5087 18.5087  75.766 9.224e-13 ***
## s(wind, 2)         1  6.8179  6.8179  27.909 1.368e-06 ***
## Residuals         70 17.1000  0.2443
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Anova for Nonparametric Effects
##                      Npar Df Npar F    Pr(F)
## (Intercept)
## s(radiation, 2)            1 0.6674 0.416747
## s(temperature, 2)          1 7.2818 0.008723 **
## s(wind, 2)                 1 3.6838 0.059022 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
gam.ozone3 = gam(ozone~s(radiation,3)+s(temperature,3)+s(wind,3), data=train)
```

```
## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts argument
## ignored
```

```r
par(mfrow=c(1,3))
plot.Gam(gam.ozone3 ,col="blue",se=T)
```




```r
summary(gam.ozone3)
```

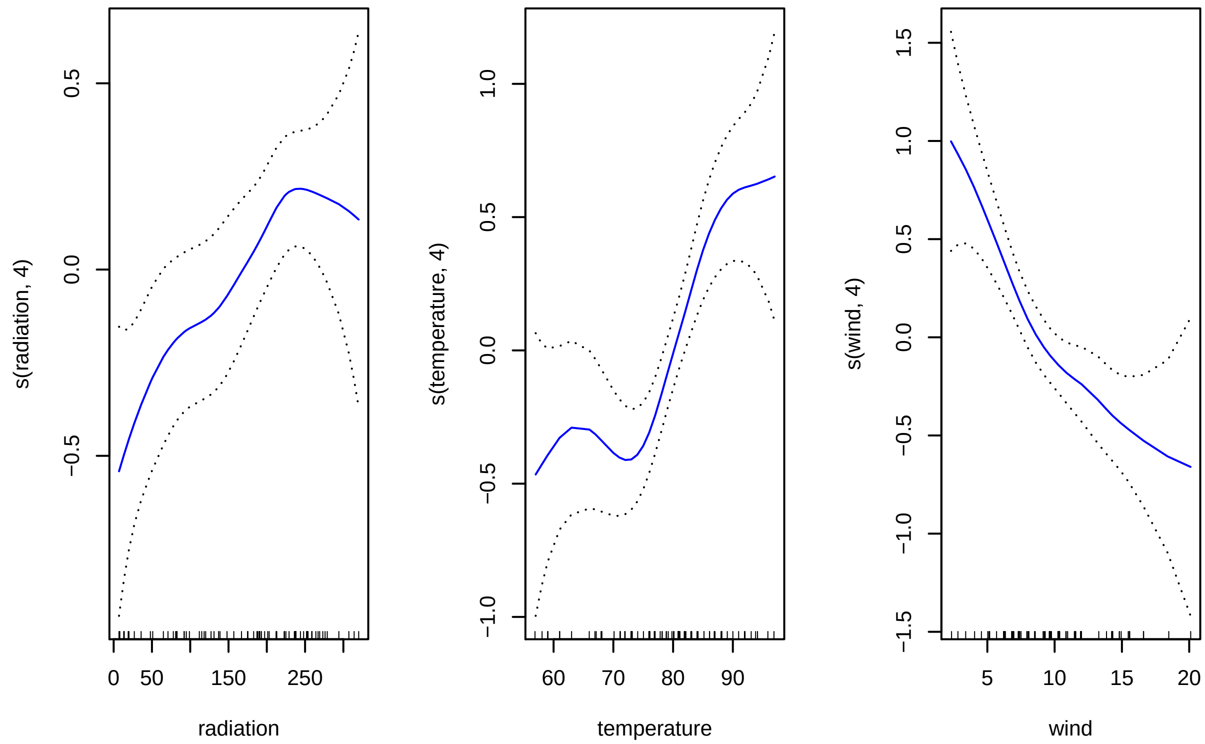
```
##
## Call: gam(formula = ozone ~ s(radiation, 3) + s(temperature, 3) + s(wind,
##     3), data = train)
## Deviance Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -1.33924 -0.24944 -0.05127  0.33862  1.12667
##
## (Dispersion Parameter for gaussian family taken to be 0.2346)
##
##     Null Deviance: 59.8074 on 76 degrees of freedom
## Residual Deviance: 15.7198 on 67.0001 degrees of freedom
## AIC: 118.1724
##
## Number of Local Scoring Iterations: 2
##
## Anova for Parametric Effects
##                   Df  Sum Sq Mean Sq F value    Pr(>F)
## s(radiation, 3)    1 13.8453 13.8453  59.011 9.016e-11 ***
## s(temperature, 3)  1 17.8958 17.8958  76.274 1.150e-12 ***
## s(wind, 3)         1  6.7093  6.7093  28.596 1.160e-06 ***
## Residuals         67 15.7198  0.2346
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Anova for Nonparametric Effects
##                   Npar Df Npar F    Pr(F)
## (Intercept)
## s(radiation, 3)         2 0.7668 0.468539
## s(temperature, 3)       2 5.6568 0.005369 **
## s(wind, 3)              2 2.0605 0.135392
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
gam.ozone4 = gam(ozone~s(radiation,4)+s(temperature,4)+s(wind,4), data=train)
```

```
## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts argument
## ignored
```

```r
par(mfrow=c(1,3))
plot.Gam(gam.ozone4 ,col="blue",se=T)
```




```r
summary(gam.ozone4)
```

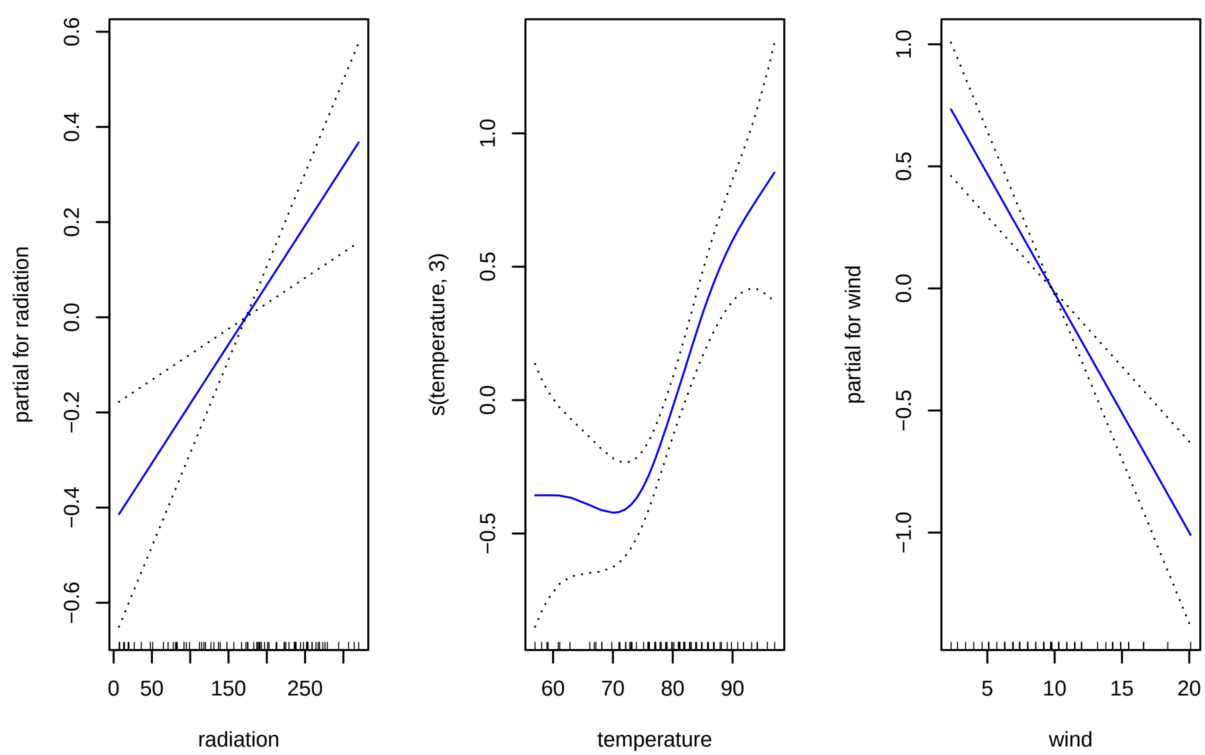
```
##
## Call: gam(formula = ozone ~ s(radiation, 4) + s(temperature, 4) + s(wind,
##     4), data = train)
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.28687 -0.22476 -0.04568  0.30784  1.06967
##
## (Dispersion Parameter for gaussian family taken to be 0.2279)
##
##     Null Deviance: 59.8074 on 76 degrees of freedom
## Residual Deviance: 14.5847 on 64.0005 degrees of freedom
## AIC: 118.4005
##
## Number of Local Scoring Iterations: 2
##
## Anova for Parametric Effects
##                       Df Sum Sq Mean Sq F value    Pr(>F)
## s(radiation, 4)    1.000 13.740 13.7399  60.293 8.374e-11 ***
## s(temperature, 4)  1.000 17.701 17.7006  77.674 1.199e-12 ***
## s(wind, 4)         1.000  6.544  6.5440  28.716 1.221e-06 ***
## Residuals         64.001 14.585  0.2279
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Anova for Nonparametric Effects
##                      Npar Df Npar F    Pr(F)
## (Intercept)
## s(radiation, 4)            3 0.8745 0.459037
## s(temperature, 4)          3 4.8831 0.004047 **
## s(wind, 4)                 3 1.6058 0.196752
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
gam.ozone = gam(ozone~radiation+s(temperature,3)+wind, data=train)
```

```
## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts argument
## ignored
```

```
par(mfrow=c(1,3))
plot.Gam(gam.ozone ,col="blue",se=T)
```




```
summary(gam.ozone)
```

```
##
## Call: gam(formula = ozone ~ radiation + s(temperature, 3) + wind, data = train)
## Deviance Residuals:
##       Min       1Q   Median       3Q      Max
```

```
## -1.55181 -0.26448 -0.01244  0.29094  1.37465
##
## (Dispersion Parameter for gaussian family taken to be 0.2381)
##
##     Null Deviance: 59.8074 on 76 degrees of freedom
## Residual Deviance: 16.9076 on 70.9999 degrees of freedom
## AIC: 115.7818
##
## Number of Local Scoring Iterations: 2
##
## Anova for Parametric Effects
##                   Df  Sum Sq Mean Sq F value    Pr(>F)
## radiation          1 14.2613 14.2613  59.887 5.101e-11 ***
## s(temperature, 3)  1 19.1043 19.1043  80.224 2.821e-13 ***
## wind               1  6.8685  6.8685  28.843 9.435e-07 ***
## Residuals         71 16.9076  0.2381
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Anova for Nonparametric Effects
##                   Npar Df Npar F    Pr(F)
## (Intercept)
## radiation
## s(temperature, 3)       2 7.0413 0.001623 **
## wind
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

I used smoothing spline for each variable and tried a few degrees of freedom. It turns out that when df=3 the AIC is the smallest. Since AIC puts a penalty on the complexity of model, it begins to increase when df>=4 even though the fitness on the training data improves.

To avoid overfitting, I chose the GAM with smoothing spline applied to each variable with df=3. The test shows the parametric effect (linear part) of all three parameters is significant while the nonparametric effect (non-linear part) is only significant for temperature. This result implies we can simply use a linear function for wind and radiation as the model gam.ozone shows.

The result below is based on model gam.ozone3: ozone~s(radiation,3)+s(temperature,3)+s(wind,3)

```r
# prediction
predict0=predict(lm.ozone,test)
predict1=predict(gam.ozone3, test)
```

```r
mse0=mean((predict0-test$ozone)^2)
mse1=mean((predict1-test$ozone)^2)
mse0
```

```
## [1] 0.263807
```

```r
mse1
```

```
## [1] 0.2070116
```

The test error of linear model is 0.26 while that of gam is 0.20. The non-linear model fits better for test data, which implies that the relation between response and expanatory variables is more complicated than linearity.

From the plot (the dash line shows the confidence interval), we noticed that the relation between wind and the response variable seems to be most linear. It seems temperature and radiation somehow have a non-linear relation with concentration.

Note: the result may differ quite a lot for different seeds. This is probably due to the small size of the dataset. All the results here are obtained with seed(1).