STAT452/652 Solution to HW03 (T's CV)

Due on Oct 02, 2020

1 Concepts

1.1 Question 1

Sample size affects variance.

1.2 Question 2

Fitting a model that is too simple causes bias.

1.3 Question 3

If we drastically increase the sample size used to fit the models in Figure 2, the following would happen:

- The estimated MSPE of the linear model would not change much, because this model's MSPE is dominated by bias. It might decrease a little, because the variance of the model would go down a bit.
- The estimated MSPE of the quadratic model would decrease a little more than that of the linear model, but still not much. The quadratic model has more variability than the linear model, but the quadratic one is still biased. This bias limits the improvement that can be achieved by increasing the sample size.
- The estimated MSPE of the quartic would definitely decrease. I might call it "a lot", but it could be considered "a little". The predictions would match the true structure essentially perfectly and have essentially no variablity beyond what we get with a new set of δ's in the test set.

- The estimated MSPE of the 7th degree polynomial would decrease a lot. This model is also unbiased, but has a tremendous amount of variance in the small sample, As with the quartic, that variance would largely boil away, leaving only the variance from new δ's.

1.4 Question 4

With a very large sample size, the opportunity for overfitting is quite limited. There would be lots of data relative to the complexity of the models. Therefore, the sMSE would resemble the MSPE in each case and essentially estimate the variance of the *delta*'s. * Linear would not change much, maybe a little bit of decrease. * Quadtatic would not change much or decrease a little. * Quartic would increase a little or a lot, because it is currently overfitting due to the samll sample size. * 7th order would increase a lot, because it is badly overfitting.

2 Application

2.1 Question 1

After removing missing values, the number of observations we are left with is given by the first component of the output of the following code.

```
data("airquality")
AQ = na.omit(airquality[,1:4])
dim(AQ)
```

```
## [1] 111   4
```

2.2 Question 2

Row numbers of observations in the validation set are given by the following code.

```
set.seed(4099183)

n = nrow(AQ) # Sample size
reorder = sample.int(n) # Randomized order

### Identify which observations go in the training set (set == 1) or the
### validation set (set == 2)
prop.train = 0.75 # Proportion of observations to put in the training set
set = ifelse(test = (reorder < prop.train * n),
  yes = 1,
  no = 2)

### Print which observations go in the validation set.
print(which(set == 2))
```

```
##  [1]   1   4   8  11  12  13  15  17  18  20  22  27  28  32  35  38  41  42  50
## [20]  51  52  59  63  69  74  87  96 111
```

```
### Construct training and validation sets
data.train = AQ[set == 1, ]
data.valid = AQ[set == 2, ]
```

2.3 Question 3

Validation set MSPEs are calculated as follows.

```
### Fit regression models
fit.solar = lm(Ozone ~ Solar.R, data = data.train)
fit.wind = lm(Ozone ~ Wind, data = data.train)
fit.temp = lm(Ozone ~ Temp, data = data.train)
fit.all = lm(Ozone ~ ., data = data.train)
fit.2 = lm(Ozone ~ . ^ 2 + I(Solar.R^2) + I(Wind^2) + I(Temp^2), data = data.train)

### Create a function which gets predictions and calculates MSPEs
MSPE.lm = function(fit, data.valid, Y.valid) {
  pred = predict(fit, data.valid) # Predicted values on validation set
  errs = Y.valid - pred # Validation set residuals
```

```
  MSPE = mean(errs ^ 2)
  return(MSPE)
}

### Create container for MSPEs
all.MSPEs = rep(0, times = 5)
names(all.MSPEs) = c("Solar.R", "Wind", "Temp", "All", "Second-Order")

### Calculate MSPEs
all.MSPEs["Solar.R"] = MSPE.lm(fit.solar, data.valid, data.valid$Ozone)
all.MSPEs["Wind"] = MSPE.lm(fit.wind, data.valid, data.valid$Ozone)
all.MSPEs["Temp"] = MSPE.lm(fit.temp, data.valid, data.valid$Ozone)
all.MSPEs["All"] = MSPE.lm(fit.all, data.valid, data.valid$Ozone)
all.MSPEs["Second-Order"] = MSPE.lm(fit.2, data.valid, data.valid$Ozone)

### Round results, then print
print(round(all.MSPEs, digits = 0))
```

```
##      Solar.R         Wind         Temp          All Second-Order
##          903          541          410          263          272
```

We see that the model with second-order terms has the smallest validation set MSPE.

2.4 Question 4

```
K = 5 # Number of CV folds

### Create container for MSPEs
CV.MSPEs = array(0, dim = c(5, K))
rownames(CV.MSPEs) = names(all.MSPEs) # Copy model names from previous question

### Create function which constructs folds for CV
### n is the number of observations, K is the number of folds
get.folds = function(n, K) {
  folds = floor((sample.int(n)-1)*K/n) + 1
  return(folds)
}

### Get CV fold labels
folds = get.folds(n, K)

### Perform cross-validation
for (i in 1:K) {
  ### Get training and validation sets
  data.train = AQ[folds != i,]
  data.valid = AQ[folds == i,]

  ### Fit regression models
  fit.solar = lm(Ozone ~ Solar.R, data = data.train)
  fit.wind = lm(Ozone ~ Wind, data = data.train)
  fit.temp = lm(Ozone ~ Temp, data = data.train)
  fit.all = lm(Ozone ~ ., data = data.train)
  fit.2 = lm(Ozone ~ . ^ 2 + I(Solar.R^2) + I(Wind^2) + I(Temp^2), data =                data.train)
```

```
  ### Calculate MSPEs
  CV.MSPEs["Solar.R", i] = MSPE.lm(fit.solar, data.valid, data.valid$Ozone)
  CV.MSPEs["Wind", i] = MSPE.lm(fit.wind, data.valid, data.valid$Ozone)
  CV.MSPEs["Temp", i] = MSPE.lm(fit.temp, data.valid, data.valid$Ozone)
  CV.MSPEs["All", i] = MSPE.lm(fit.all, data.valid, data.valid$Ozone)
  CV.MSPEs["Second-Order", i] = MSPE.lm(fit.2, data.valid, data.valid$Ozone)
}

### Get mean and SD for each model's MSPE
CV.stats = apply(CV.MSPEs, 1, function(errs) {
  ### Container for summary statistics
  output = rep(0, times = 2)
  names(output) = c("mean", "SD")

  ### Calculate summary statistics
  output["mean"] = mean(errs)
  output["SD"] = sd(errs)

  return(output)
})

### Create function to calculate 95% CI based on mean, SD and sample size
### To save time, also output the mean
get.CI = function(mean, SD, n) {
  T.val = qt(0.975, n - 1)
  MOE = T.val * SD / sqrt(n)

  output = rep(0, times = 3)
  names(output) = c("mean", "lcl", "ucl")

  output["mean"] = mean
  output["lcl"] = mean - MOE
  output["ucl"] = mean + MOE

  return(output)
}

### Get CI for each model's MSPE
CV.CIs = apply(CV.stats, 2, function(info) {
  return(get.CI(info["mean"], info["SD"], K))
})

### I prefer models as rows for this output
### Don't forget to round
print(round(t(CV.CIs), digits = 0))
```

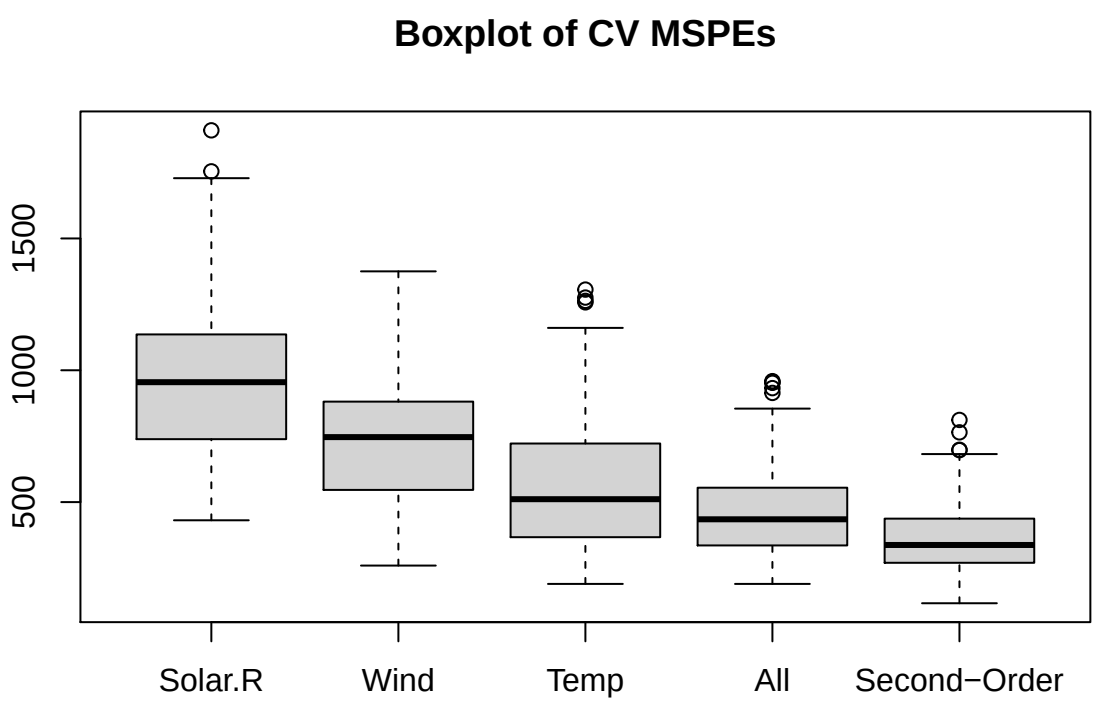
```
##              mean lcl  ucl
## Solar.R      1051 515 1587
## Wind          747 402 1091
## Temp          607 171 1043
## All           477 178  775
## Second-Order  372 198  545
```

The models with all predictors seem to be doing best; particularly the one which also includes second-order

terms. The model with only solar radiation appears to be the worst, and the other univariate models are somewhere in the middle. HOWEVER, it might be observed that ALL models have overlapping confidence intervals for their MSPEs. Just based on these figures, it is hard to make strong conclusions.

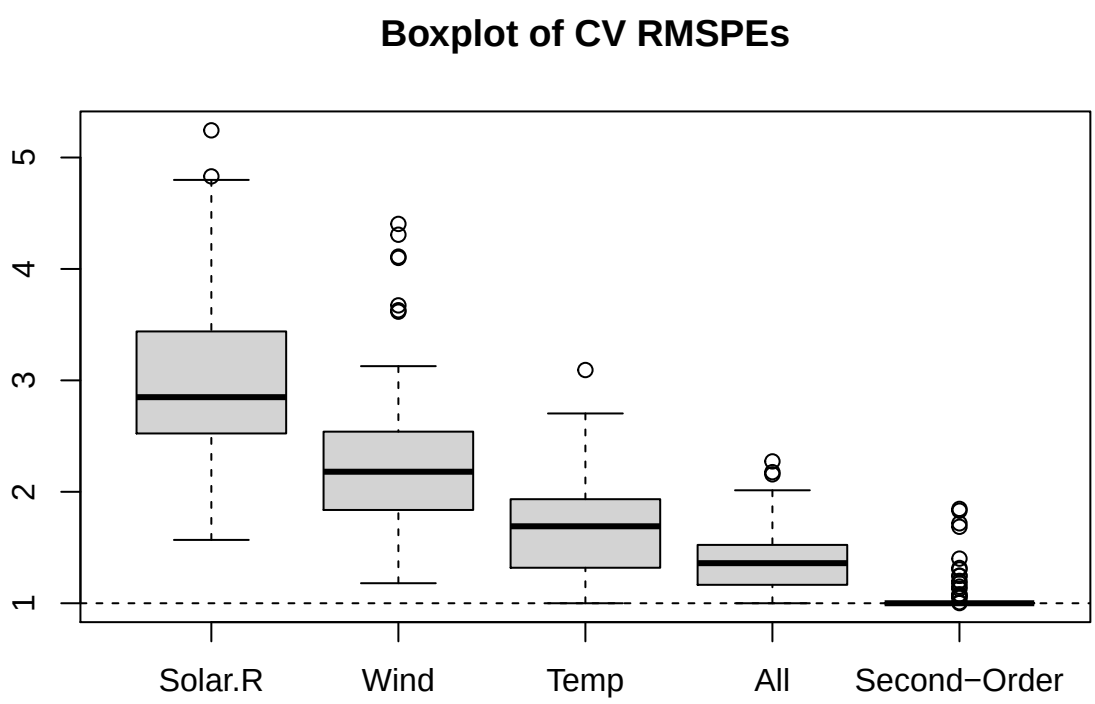
2.5 Question 5

```
M = 20 # Number of times to repeat CV

### Create a 3D array to store CV errors by model, then iteration,
### then fold. Set names of models.
all.CV.MSPEs = array(0, dim = c(5, M, K))
dimnames(all.CV.MSPEs)[[1]] = rownames(CV.MSPEs)


for (j in 1:M) {
  ### Get CV fold labels
  folds = get.folds(n, K)

  ### Perform cross-validation
  for (i in 1:K) {
    ### Get training and validation sets
    data.train = AQ[folds != i,]
    data.valid = AQ[folds == i,]

    ### Fit regression models
    fit.solar = lm(Ozone ~ Solar.R, data = data.train)
    fit.wind = lm(Ozone ~ Wind, data = data.train)
    fit.temp = lm(Ozone ~ Temp, data = data.train)
    fit.all = lm(Ozone ~ ., data = data.train)
    fit.2 = lm(Ozone ~ . ^ 2 + I(Solar.R^2) + I(Wind^2) + I(Temp^2), data = data.train)

    ### Calculate MSPEs
    ### Be careful with order of indices!!!
    all.CV.MSPEs["Solar.R", j, i] = MSPE.lm(fit.solar, data.valid, data.valid$Ozone)
    all.CV.MSPEs["Wind", j, i] = MSPE.lm(fit.wind, data.valid, data.valid$Ozone)
    all.CV.MSPEs["Temp", j, i] = MSPE.lm(fit.temp, data.valid, data.valid$Ozone)
    all.CV.MSPEs["All", j, i] = MSPE.lm(fit.all, data.valid, data.valid$Ozone)
    all.CV.MSPEs["Second-Order", j, i] = MSPE.lm(fit.2, data.valid, data.valid$Ozone)
  }
}

### Combine all MSPEs for each model into a single list
### Note: as.numeric() converts its input into a 1D vector
data.CV.MSPEs = apply(all.CV.MSPEs, 1, as.numeric)

### Create boxplot
boxplot(data.CV.MSPEs, main = "Boxplot of CV MSPEs")
```

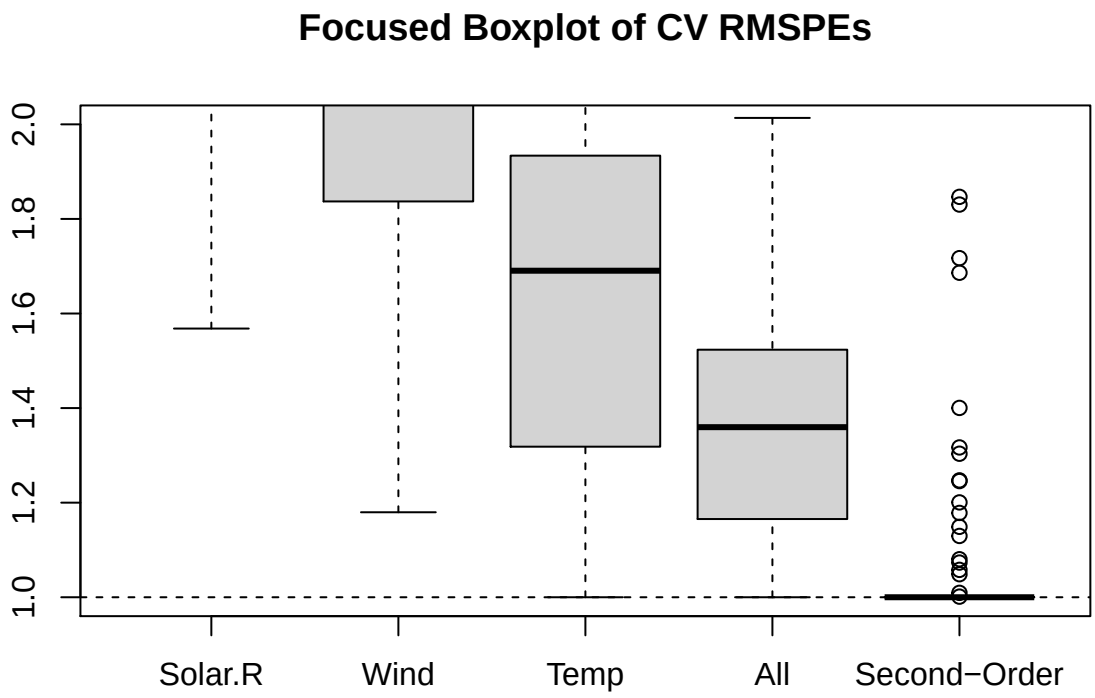


This boxplot of CV-MSPEs suggests that the full and second-order models are best, although not by much. The model with just Temp has just slightly higher box and a little more upward spread. The models with just Wind and just Solar.R appear progressively worse, but all of the models' boxplots overlap a considerable amount.

```
### Compute RMSPEs
data.CV.RMSPEs = apply(data.CV.MSPEs, 1, function(W){
  best = min(W)
  output = W / best
  return(output)
})
data.CV.RMSPEs = t(data.CV.RMSPEs)

### Create boxplot
boxplot(data.CV.RMSPEs, main = "Boxplot of CV RMSPEs")
abline(h = 1, lty = 2)
```




```
### Create boxplot
boxplot(data.CV.RMSPEs, main = "Focused Boxplot of CV RMSPEs", ylim = c(1,2))
abline(h = 1, lty = 2)
```



The second-order model is best most often. It is sometimes beaten, but not often. The only other models that ever win the competition are the full model, and the model containing only Temp. This plot also more clearly shows how much worse each other model is in general.

2.6 Question 6

The model with all main effects and second-order interations consistently performed best across many data splits. This tells us that, although Solar.R is difficult to measure, having all variables is important. If some performance can be sacrificed in favor of saving effort on measurement, Temp is the best to include if you can only choose one predictor. But you can expect the MSPE to increase by around 50% or more as a result.