Intro to ML PS 1

Takahiro Minami

1/17/2020

Question 1

In supervised machine learning, the predictor feature measurements, Xs, are associated with response measurement, Y, and we assume that we can express or predict Y as a function of Xs. What we try to do in supervised machine learning is to estimate this inferred function based on observed pairs of outcomes Y and inputs Xs, which are called "training data" in the machine learning world. To be more specific, we are supposed to select an optimal model among many potential hypothetical models based on some evaluation and optimization methodologies (e.g. cross validation). Here, we need to know how the data of Y and Xs are generated and how the data is structured because we have to build hypothetical models about the relationship between inputs and outcome. Once we estimate the inferred function, we can predict unobserved outcome measure by using input measures, which is called "test data," and the function. This is our final goal to conduct supervised machine learning.

Typical examples of supervised machine learning are classification and regression. classification method will be used for binary or categorical measurement variables Y, while regression method will be used for continuous measurement variables. In these methods, we estimate the function which can assign individuals with characteristics Xs to correct classes Y, or can predict Y precisely based on Xs.

On the other hand, unsupervised machine learning is importantly different from supervised learning. The most essential difference is that there are no outcome measures associated with input measures in unsupervised machine learning. In other words, the data of Xs are not labeled. While pairs of Y and Xs in training sample can teach us about optimal relationship between Y and Xs in supervised learning, there is no such a "teacher" in unsupervised learning, and we have only Xs. Furthermore, in unsupervised learning, we don't know how the data are generated nor the structure of data. Given these situation, our goal of unsupervised machine learning is to figure out how Xs are organized by examining and finding out various characteristics hidden in the data. This means that unsupervised learning is not so useful to predict some measurements. Instead, it is for discovering data structures inside the data.

Typical examples of unsupervised machine learning are clustering and association. In both cases, the machine learning algorithms analyze various characteristics among the data and categorize them (clustering) or find out connection between some group of data and others (association). The point here is that these categories and relationships cannot be pre-defined based on past experiences or observation, which is consistent with the fact that there are no teachers in unsupervised learning. The algorithms find out them from complex data by themselves. \

To clarify the differences between supervised and unsupervised machine learning mentioned above, suppose we have various kind of vegetables in front of us. In supervised learning, we need to know the name of vegetables (Y) and their characteristics including shape, color, size, and so on (Xs). By using classification method, we will estimate the function to predict class (vegetable name) based on various characteristics. Once we get this function, we can guess which vegetable they are when we get new vegetables. On the other hand, in unsupervised machine learning, we don't know name of vegetable nor how each vegetable (e.g. tomato) look like. The clustering algorithms examine shape, color and size of these unknown vegetables and make groups (clusters) based on these characteristics.

Question 2

```
df <- mtcars
names(df)
```

```
##  [1] "mpg"  "cyl"  "disp" "hp"   "drat" "wt"   "qsec" "vs"   "am"   "gear"
## [11] "carb"
```

a.

To predict miles per gallon by cylinders, I regress *mpg* on *cyl*. I regard *mpg* as Y and *cyl* as X in this case. Here is the output and estimated parameters.

```
fit.lm <- lm(mpg ~ cyl, data=df)
summary(fit.lm)
```

```
##
## Call:
## lm(formula = mpg ~ cyl, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.9814 -2.1185  0.2217  1.0717  7.5186
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  37.8846     2.0738   18.27  < 2e-16 ***
## cyl          -2.8758     0.3224   -8.92 6.11e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.206 on 30 degrees of freedom
## Multiple R-squared:  0.7262, Adjusted R-squared:  0.7171
## F-statistic: 79.56 on 1 and 30 DF,  p-value: 6.113e-10
```

```
paste0("The constant is ",round(coef(fit.lm)[[1]],2),".")
```

```
## [1] "The constant is 37.88."
```

```
paste0("The parameter on cyl is ",round(coef(fit.lm)[[2]],2),".")
```

```
## [1] "The parameter on cyl is -2.88."
```

b.

The population regression function is

$$mpg_i = \beta_0 + \beta_1 cyl_i + \epsilon_i$$

c.

I run the following regression.

$$mpg_i = \beta_0 + \beta_1 cyl_i + \beta_2 wt_i + \epsilon_i$$

```r
fit.lm2 <- lm(mpg ~ cyl + wt, data=df)
summary(fit.lm2)
```

```
##
## Call:
## lm(formula = mpg ~ cyl + wt, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2893 -1.5512 -0.4684  1.5743  6.1004
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  39.6863     1.7150  23.141  < 2e-16 ***
## cyl          -1.5078     0.4147  -3.636 0.001064 **
## wt           -3.1910     0.7569  -4.216 0.000222 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.568 on 29 degrees of freedom
## Multiple R-squared:  0.8302, Adjusted R-squared:  0.8185
## F-statistic: 70.91 on 2 and 29 DF,  p-value: 6.809e-12
```

```r
paste0("The constant is ",round(coef(fit.lm2)[[1]],2),".")
```

```
## [1] "The constant is 39.69."
```

```r
paste0("The parameter on cyl is ",round(coef(fit.lm2)[[2]],2),".")
```

```
## [1] "The parameter on cyl is -1.51."
```

```r
paste0("The parameter on wt is ",round(coef(fit.lm2)[[3]],2),".")
```

```
## [1] "The parameter on wt is -3.19."
```

Comparing the result in question (a) and (c), constant is almost the same. However, the parameter on *cyl* variable (absolute value) becomes small once I add *wt* variable to the regression. The parameter on *cyl* variable means the effect of increase of number of cylinder by 1 on miles per gallon. Therefore, based on the result of question (c), I can interpret that the distance a car can run consuming 1 gallon of gas will be reduced by 1.51 miles when the number of cylinder increases by 1. Because in the model in question (a), the reduction of miles per gallon associated with 1 more cylinder is 2.88, the model in this question implies that the effect of number of cylinder on miles per gallon is smaller than I found in question (a).

d.

I run the following regression.

$$mpg_i = \beta_0 + \beta_1 cyl_i + \beta_2 wt_i + \beta_3 cyl_i * wt_i + \epsilon_i$$

```r
fit.lm3 <- lm(mpg ~ cyl + wt + I(cyl*wt), data=df)
summary(fit.lm3)
```

```
##
## Call:
## lm(formula = mpg ~ cyl + wt + I(cyl * wt), data = df)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2288 -1.3495 -0.5042  1.4647  5.2344
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  54.3068     6.1275   8.863 1.29e-09 ***
## cyl          -3.8032     1.0050  -3.784 0.000747 ***
## wt           -8.6556     2.3201  -3.731 0.000861 ***
## I(cyl * wt)   0.8084     0.3273   2.470 0.019882 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.368 on 28 degrees of freedom
## Multiple R-squared:  0.8606, Adjusted R-squared:  0.8457
## F-statistic: 57.62 on 3 and 28 DF,  p-value: 4.231e-12
```

```
paste0("The constant is ",round(coef(fit.lm3)[[1]],2),".")
```

```
## [1] "The constant is 54.31."
```

```
paste0("The parameter on cyl is ",round(coef(fit.lm3)[[2]],2),".")
```

```
## [1] "The parameter on cyl is -3.8."
```

```
paste0("The parameter on wt is ",round(coef(fit.lm3)[[3]],2),".")
```

```
## [1] "The parameter on wt is -8.66."
```

```
paste0("The parameter on interaction is ",round(coef(fit.lm3)[[4]],2),".")
```

```
## [1] "The parameter on interaction is 0.81."
```

Even after I add interaction term, the sign of *cyl* and *wt* variables are still negative, which are consistent with the result in question (c). This means that increase of number of cylinder and the weight of a car are associated with reduction of distance a car can run consuming 1 gallon. However, the absolute value of both parameters become larger in the model with interaction term. \ Theoretically, interaction term tries to measure how increase of weight affects the effect of cylinder on miles per gallon, or how increase of cylinder affects the effect of weight on miles per gallon. In other words, it measures the effect of cylinder (or weight) on the slope between miles per gallon (y) and weight (or cylinder)(x). In this case, the positive parameter of interaction term implies that the negative effect of increasing number of cylinder (weight) is mitigated when weight (or number of cylinder) increases.

Question 3

```
df_w <- read.csv("wage_data.csv", header = TRUE)
head(df_w)
```

```
##        X year age           maritl     race       education
## 1 231655 2006  18 1. Never Married 1. White    1. < HS Grad
## 2  86582 2004  24 1. Never Married 1. White 4. College Grad
## 3 161300 2003  45       2. Married 1. White 3. Some College
## 4 155159 2003  43       2. Married 3. Asian 4. College Grad
## 5  11443 2005  50      4. Divorced 1. White      2. HS Grad
## 6 376662 2008  54       2. Married 1. White 4. College Grad
##               region       jobclass         health health_ins  logwage
## 1 2. Middle Atlantic  1. Industrial      1. <=Good      2. No 4.318063
```

```
## 2 2. Middle Atlantic 2. Information 2. >=Very Good      2. No 4.255273
## 3 2. Middle Atlantic  1. Industrial      1. <=Good     1. Yes 4.875061
## 4 2. Middle Atlantic 2. Information 2. >=Very Good     1. Yes 5.041393
## 5 2. Middle Atlantic 2. Information      1. <=Good     1. Yes 4.318063
## 6 2. Middle Atlantic 2. Information 2. >=Very Good     1. Yes 4.845098
##        wage
## 1  75.04315
## 2  70.47602
## 3 130.98218
## 4 154.68529
## 5  75.04315
## 6 127.11574
```

a.

I run the following regression.

$$wage_i = \beta_0 + \beta_1 age_i + \beta_2 (age_i)^2 + \epsilon_i$$

```
fit.lm4 <- lm(wage ~ age + I(age^2) , data=df_w)
summary(fit.lm4)
```

```
##
## Call:
## lm(formula = wage ~ age + I(age^2), data = df_w)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -99.126 -24.309  -5.017  15.494 205.621
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -10.425224   8.189780  -1.273    0.203
## age           5.294030   0.388689  13.620   <2e-16 ***
## I(age^2)     -0.053005   0.004432 -11.960   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 39.99 on 2997 degrees of freedom
## Multiple R-squared:  0.08209,    Adjusted R-squared:  0.08147
## F-statistic:   134 on 2 and 2997 DF,  p-value: < 2.2e-16
```

```
paste0("The constant is ",round(coef(fit.lm4)[[1]],2),".")
```

```
## [1] "The constant is -10.43."
```

```
paste0("The parameter on age is ",round(coef(fit.lm4)[[2]],2),".")
```

```
## [1] "The parameter on age is 5.29."
```

```
paste0("The parameter on age^2 is ",round(coef(fit.lm4)[[3]],2),".")
```

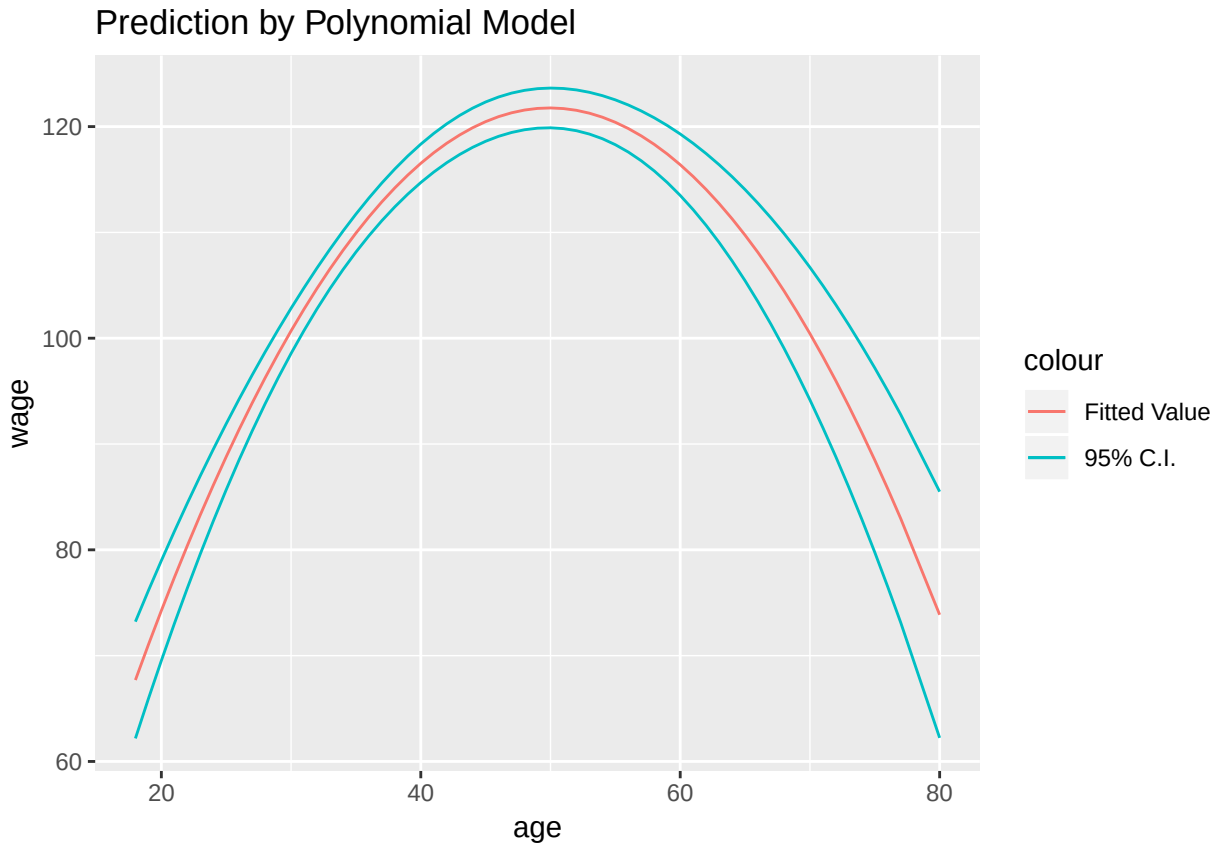
```
## [1] "The parameter on age^2 is -0.05."
```

Looking at the result, parameters on both age and age^2 are statistically significant at 1% level, which means that the 2nd order polynomial term of age has significant explanation power for the relationship between wage

and age. Although the negative intercept cannot be interpreted reasonably, it doesn't have much meaning because it's meaningless to predict wage at age of zero and the intercept is not statisticaly significant.

b.

```
df_predict <-
  as.data.frame(predict(fit.lm4,newdata=df_w, interval = "confidence"))%>%
              cbind(df_w$age)
colnames(df_predict)<- c("fit","lower","upper","age")
ggplot()+
  geom_line(data= df_predict, aes(x = age, y = fit, , col="black")) +
  geom_line(data= df_predict, aes(x = age, y = lower,  col="blue")) +
  geom_line(data= df_predict, aes(x = age, y = upper,  col="blue")) +
  labs(title = "Prediction by Polynomial Model",
       x = "age", y = "wage") +
  scale_color_discrete(labels = c("Fitted Value", "95% C.I."))
```



c.

Since the parameter of age^2 is negative, the estimated function is concave, and marginal effect of increasing age on wage is decreasing. The impact of getting old on wage is larger for young people than older people.\ The function take global maximum at around age of 50. I can interpret that wage increasing as people get older since the age of 50, but after 50, getting older is associated with decrease of wage. \ These complex relationship between age and wage cannot be interpreted by simple linear model. Then, finding these relationship is clearly one of the biggest benefit to use polynomial model.

d.

Polynomial model have higher order of variable x, x^2, x^3, and so on, in its independent variables while linear model has only the first order of x in right hand side. Polynomial regression has more flexibility to fit the model to the observed data because polynomial model can generate non-linear, complicated shape of fitted line to minimize RSS. However, this advantage can be disadvantage. Polynomial regression can be easily affected by outlier in data because polynomial model is able to fit the line even to the outlier although it doesn't or shouldn't do so actually. This problem can be generalized as over-fitting problem. Because of the flexibility of fitting, polynomial model can be fitted very much to training sample, but it's possible that the model has very bad performance when it is applied to general data (test sample). \ In addition to these statistical characteristics, there are some substantive characteristics. As seen in question (c), polynomial model enables us to find non-linear relationship between y and x. In real world, most of variables have more complicated relationship than linear one. So, polynomial model is more useful to capture these relationship (e.g. change of marginal impact of x on y), while simple linear model may too simplified the relationships and mislead us. On the other hand, polynomial model has substantive pitfalls, too. If we pursue only the best fit of the model, the estimated parameters of higher orders are difficult to be interpreted. For example, we cannot interpret the meaning of parameter on x^{20}. This problem is similar to the over-fitting problem, and suggests that polynomial with too many orders is sometimes worse than simple linear model.