PS1

Joanna Zhang

1/17/2020

Statistical and Machine Learning

Supervised machine learning aims at uncovering the true population function using the training data, where all observations in the data set are labeled and the features of the function are known. In supervised machine learning, we assume that we have a correct understanding of the predictors in the true function (i.e. we know what variables are in the true function).One can use supervised learning to solve classifications and regressions. Unsupervised machine learning aims at finding patterns in the data. In this case, we do not have knowledge about the true population function. We let the learning algorithm explore and present the patterns in the data. Unsupervised learning is widely used for clustering problems and dimension reduction problems.

What is the relationship between the X's and Y? In supervised learning, X's are associated with Y. We observe both the X's and the Y. We try to use the learning algorithm to find the function that can represent this association between X's and Y. While in unsupervised learning, we only observe the X's and the Y us unknown. There is no predetermined Y that is associated with the X's. We let the learning algorithm find the possible Y's.

What is the target we are interested in? For supervised learning, we want to find the parameters in the true population function based on the training data and be as precise as possible. For unsupervised learning, we are interested in finding the inherent structure of the data.

How do we think about data generating processes? In supervised learning, we may assume that the true population function that describes the relationship between X's and Y is already embedded in the data generating process. In generating the data, X's and Y follow a certain distribution, and this distribution can be derived from the true population function. For unsupervised learning, although we do not know what the population function looks like, we still think that there is some inherent structure that connects X's to a Y, which is unknown. We believe that the data is generated based on the inherent structure, but we do not have information about the structure before running the learning algorithm.

What are our goals in approaching data? In supervised learning, our goal is to find the population function that describes the relationship between the X's and the Y's so that we can make accurate out-of-sample predictions. With the function, in some other scenarios when we can only observe X's, we may still predict the value of Y. The goal in unsupervised learning is to uncover the structures and patterns in the data that could be useful for future analysis.

How is learning conceptualized? For supervised learning, we think of leaning as producing estimations of population parameters and conditional distribution using the training data. In unsupervised leaning, the concept of learning means that the algorithm would explore the data without instruction. The algorithm tries to understand the structure of the data and produces outputs that could be the underlying structure of the data.

Linear Regression

a)

```
# load the data
data(mtcars)
# run the regression with single independent var
mpgmodel <- lm(mpg ~ cyl, mtcars)
```

```r
# get the summary info of the model
summary(mpgmodel)
```

```
##
## Call:
## lm(formula = mpg ~ cyl, data = mtcars)
##
## 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
```

The regression output is listed above. From the summary we see that the estimation for intercept parameter is 37.88 and the estimation for the coefficient of cylinders is -2.88. Both the parameters has a small p-value, meaning that they are statistically significant. For one unit increase in cylinders, the mpg is expected to drop by 2.88. And we have an R-squared of 0.73, meaning that we are explaining 73% of the variance in MPG.

b)

The statistical form of the model is $\widehat{MPG}_i = 37.88 - 2.88\ \widehat{Cylinders}_i + \epsilon_i$.

c)

```r
# run the regression with two explanatory variables
mpgmodel_wt <- lm(mpg ~ cyl + wt, mtcars)
summary(mpgmodel_wt)
```

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

After adding weight to the model, we see that weight has a coefficient of -3.19, which means that for one unit increase in the weight of a car, on average, the MPG reduces by 3.19 unites, holding the number of cylinders constant. As we add weight to the model, the coefficient of cylinders becomes less negative (and smaller in absolute value). Now with weight constant, as the number of cylinders increases by one unit, the MPG is expected to drop by 1.5 units. Note that both cylinders and weight have small p-values, and thus they are statistically significant in this model. As we include weight in this model, the R-squared increases to 0.83.

d)

```r
#run the regression
mpgmodel_interact <- lm(mpg ~ cyl + wt + cyl*wt, mtcars)
summary(mpgmodel_interact)
```

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

In the regression with interaction, we assume that the effect of cylinder and the effect of weight are not independent of one another. We think that the effect of increasing one unit of cylinder/weight would depend on the magnitude of weight/cylinder.

Note that the coefficients of cylinder and weight remain negative, but the magnitude of these two coefficients has become greater in absolute value.. The coefficient of cylinder means that if the weight of a car equals zero, for every unit of increase in cylinder, the expected MPG decreases by 3.8. (Although there's no car with zero weight.) Similarly, the coefficient of weight means that when the number of cylinder is zero, the expected MPG drops by 8.66 when weight increases by one unit. The interaction term has a coefficient of 0.81. The positiveness of the coefficient of the interaction implies that if the number of cylinder increases by one, the MPG of heavier cars decreases by a smaller amount than the MPG of lighter cars. Similarly, if the weight of a car increases by one unit, cars with more cylinders would see a smaller drop if the MPG than cars with less cylinders would.

Non-linear Regression

a)

```
#read the csv file
wagedata <- read_csv("wage_data.csv")
#run the regression
wage_poly_mod <- lm(wage ~ poly(age, degree = 2, raw = T), wagedata)
summary(wage_poly_mod)
```

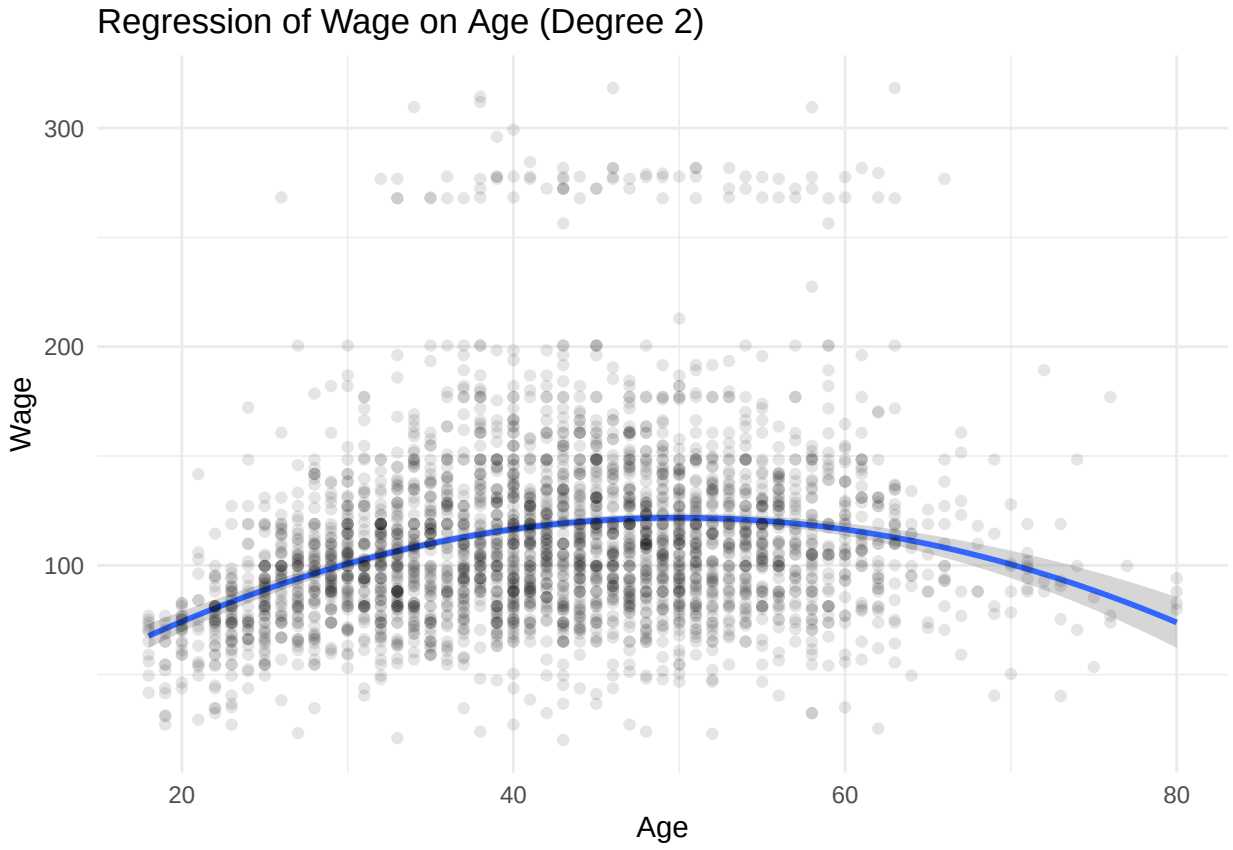
```
##
## Call:
## lm(formula = wage ~ poly(age, degree = 2, raw = T), data = wagedata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -99.126 -24.309  -5.017  15.494 205.621
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    -10.425224   8.189780  -1.273    0.203
## poly(age, degree = 2, raw = T)1   5.294030   0.388689  13.620   <2e-16 ***
## poly(age, degree = 2, raw = T)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
```

The summary above lists the regression results. The age variable has a positive coefficient of 5.29, while the squared variable has a negative coefficient. Both age and age squared has small p-values, meaning that they are both statistically significant. As age changes, the effect of age on wage changes as well. Also note that the R-squared of this model is 0.08, meaning that our model only explains 8% of the variance in wage.

b)

```
#create small intervals of age
ageseq <- seq(min(wagedata$age), max(wagedata$age), length.out=200)
#create a dataframe for predicted values
wage_predicted <- data.frame(age = ageseq)
predictedvalues <- predict(wage_poly_mod, newdata = wage_predicted, se.fit = TRUE)
#Calculate confidence interval boundaries
wage_predicted $lbd <- predictedvalues$fit - 1.96 * predictedvalues$se.fit
wage_predicted $fit <- predictedvalues$fit
wage_predicted $ubd <- predictedvalues$fit + 1.96 * predictedvalues$se.fit
#plot the confidence interval and the fitted line
ggplot(wage_predicted, mapping = aes(x = ageseq, y = fit)) +
  geom_line(color = 'blue') +
  geom_ribbon(aes(ymin = lbd, ymax = ubd), stat = "identity", alpha = 0.1)+
  theme_minimal()+
  labs(x = "Age", y = "Wage", title = "Regression of Wage on Age (Degree 2)")
```


```
#Alternatively, we can specify formula in geom_smooth and avoid calculating fitted values
wagedata %>%
  ggplot(aes(age, wage))+
  #plot the regression line with the confidence interval
  geom_smooth(method = lm,
              formula = y ~ poly(x, 2, raw = T),
              level = 0.95)+
  #plot the data points
  geom_point(alpha = 0.1)+
  theme_minimal()+
  labs(x = "Age", y = "Wage", title = "Regression of Wage on Age (Degree 2)")
```



c)

By fitting a polynomial regression, we assert that the relationship between the two variables is non-linear. We think that it is possible that the effect of a change in the independent variable may not be consistent at all values. In particular, since we use polynomial regression of degree two, we assume that the relationship should look like a parabola (symmetric with one peak). From the plot we see that the expected wage increases as age increases when age is smaller than roughly 50. When age is around 50, the predicted wage is the highest. Then as age increases, the expected wage gradually falls.

d)

Polynomial regression and linear regression have different assumptions. Linear regression assumes that the effect of each unit of the independent variable is identical no matter what value the independent variable takes. As we can see this from the general equation for linear regression:

$\hat{Y} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ...$

The effect of each unit of X_1 is identical for all X_1.For example, the relationship of someone's electricity bill and his/her electricity usage can be mapped with a linear regression, since we believe that the price of electricity is constant.

In polynomial regression, we assume that the effect of independent variable varies across different values.

$\hat{Y} = \beta_0 + \beta_1 X_1 + \beta_2 {X_1}^2 + ...$

The effect of one unit's change in X_1 differs as X_1 takes different values. The relationship of age and wage is an example here. We think that wage would increase with age because older people have more experiences at the beginning. But when age is above 50, older people might not have that much advantage compared to

younger individuals in the job market. For different ages, we want to make sure we can observe the differential effect from the model, so we use polynomial regression. Depending on the assumptions we put on our data, we can choose which type of regression to run accordingly.