PS1

Joanna Zhang

1/15/2020

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 finding association rules.

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 distrbution 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 senarios 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)

```
data(mtcars)
mpgmodel <- lm(mpg ~ cyl, mtcars)
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
```

b)

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

c)

```
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
```

d)

```
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
```

Non-linear Regression

a)

```
wagedata <- read_csv("wage_data.csv")
wage_poly_mod <- lm(wage ~ poly(age, degree = 2), wagedata)
summary(wage_poly_mod)
```

```
##
## Call:
## lm(formula = wage ~ poly(age, degree = 2), 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)             111.7036     0.7302  152.99   <2e-16 ***
## poly(age, degree = 2)1  447.0679    39.9926   11.18   <2e-16 ***
## poly(age, degree = 2)2 -478.3158    39.9926  -11.96   <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
```

b)

```
wagedata %>%
  ggplot(aes(age, wage))+
  geom_smooth(method = lm, formula = y ~ poly(x, 2))+
  theme_minimal()
```