Jiewen_Luo_ps1

Jiewen

1/17/2020

1. Statistical and Machine Learning

Supervised machine learning is an algorithm where a mathematical model is built from a set of labeled data contain both inputs Xs and the desired output Y. In supervised machine learning, the system needed to be trained to perform certain tasks. Unsupervised machine learning is a self-learning process. Under this system, an algorithm is given only the unlabeled input data Xs without corresponding output variables.

By building an appropriate model that establishes the corresponding relationship between Xs and Y, supervised learning can predict output data Y when new Xs in the data set is introduced. To accomplish this task, the algorithm iteratively predicts the training data until a given level of predicting accuracy is achieved. As for unsupervised learning, the purpose is to summarize and explain the underlying distribution or structure of the data.

The actual data generating processes in supervise learning also differ from unsupervised learning. To solve any given supervised learning problem, we first need to determine the types of data needed. The next step is to collect representative training data that well-represented the entire population. Afterward, the input feature representation of the learned function should be well-selected. A good feature vector should contain enough information for predicting output but not too large because of the course of dimensionality. Another critical step is to select the structure of the learning function and learning algorithm. Finally, the optimal model can establish after iteratively running the learning algorithm on the collected training data. For unsupervised learning, the training step is eliminated so the system has to dig in hidden patterns of the data on its own.

In terms of the methods applied, these two types of machine learning systems also distinct from each other. When dealing with continuous quantitative data, supervised learning performs regression analysis while unsupervised learning use dimensionality. When discrete data or qualitative data are given, supervised learning handles the task by classification whereas unsupervised learning adopts cluster analysis.

Here are two machine learning tasks that help to illustrate the difference between these two algorithm systems. When dealing with the grouping issue, supervised learning applies a classification algorithm. Because the data given come with categories, the task for the system is to accurately recognize category membership of each observation in the training set. After the training, the system is then able to identify the categories of new observations. In the unsupervised case, the categories of the data are unknown. Therefore, cluster analysis is used to segment data into groups. This goal can be achieved by identifying the degree of inherent similarity in the data based on the presence or absence of various indicators. In the task of density estimation, the difference can also be presented mathematically. While supervised learning aims at inferring a conditional probability distribution conditioned on the label x of input data, unsupervised learning aims at inferring a priori probability distribution.

Finally, from an evaluation perspective, these two algorithms vary in terms of complexity and accuracy. Supervised learning is a simpler method compared to unsupervised learning. In supervised learning, because the model built by supervised learning is repeatedly examined by the training data set until it reaches a target level of correctness within the training data, the result is trustworthy. However, there is no such equivalent model optimization process in unsupervised machine learning, therefore, the result is more doubtful.

2. Linear Regression

a.

```
mc=mtcars
Predict1=lm(mpg~cyl,data=mc)
summary(Predict1)

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

Output value: Regression report is shown above. F test shows that the model is statistically significant. R squared is high, suggesting that the model is quite a good fit.

Constant term is `37.8846`, and the parameter value for **cylinders** is statistically significant with value of -2.8758.

b. fitted value for this model is $\widehat{mpg}$=37.8846-2.8758*cyl; The statistical form is mpg_i = β0 +β1$*cyl_i$ + ϵ_i

c.

```
Predict2=lm(mpg~cyl+wt,data=mc)
summary(Predict2)
```

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

Now $\widehat{mpg}$2=39.6863-1.5078*cyl-3.1910*wt.
By adding **vehicle weight** into the model, the coefficient size of the constant term increases but the coefficient size for variable **cylinders** decreases. This means that the negative effect of **cylinders** on **miles per gallon** decreases. Additionally, this model also shows us that **vehicle weight** has statically and substantively significant negative effect on **miles per gallon**. Increase in 1 unit of **vehicle weight** will lead to decrease of mile per gallon by 3.1910 unit.

d.

```
mc=mtcars
Predict3=lm(mpg~cyl+wt+cyl:wt,data=mc)
summary(Predict3)
```

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

$\widehat{mpg}$3= 54.3068-3.8032*cyl-8.6556*wt + 0.8084 wt*cyl.

After adding interaction term, the joint effect of these three variables is still statistically significant. R-squared in these two models are both greater than 0.8, which means the level of goodness of fit are high. We can also see that the intercept term remains positive, and the estimates for **cylinders** and **vehicle wage** remain negative. However, the magnitude of the estiamator for **cylinders** and **vehicle wage** both increase.
By adding interaction term, we are theoretically asserting that **vehicle wage** influences the relationship between **cylinders** and **miles per gallon**.

3. Non-linear Regression

a.

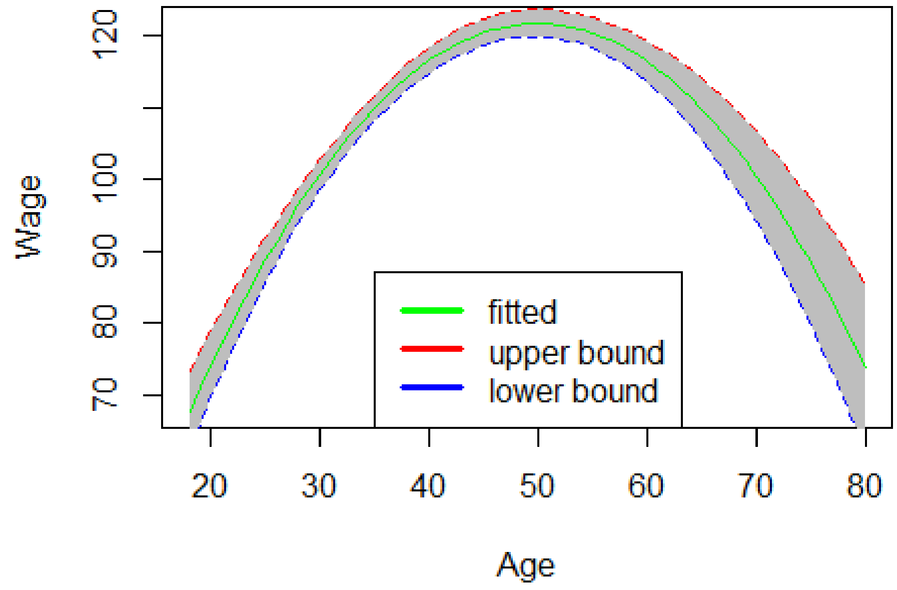
```
dt=read.csv("wage_data.csv")
model1<-lm(wage~poly(dt$age,2,raw=TRUE),data=dt)
summary(model1)

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

$\widehat{wage}1=-10.425224+5.294030*age-0.053005 *age^2$

The model is statistically significant, but has poor goodness of fit. The linear age term has statistically and substantively significant positive value and the quadratic age term has statistically and substantively significant negative value. This mean that age has diminishing marginal effect on wage. The marginal effect is positive at the begining, but as ages increases to 5.294030/(2*0.053005)=49.95, the marginal effect will diminish to 0 and then become negative afterward.

b.

```
plot(sort(dt$age), fitted(model1)[order(dt$age)],xlab="Age",ylab="Wage", col=
"green", type='l')
preds <- predict(model1, data.frame(x=dt$age), interval = "confidence",level=
0.95)
x<-sort(dt$age)
y1<-preds[ ,3][order(dt$age)]
y2<-preds[ ,2][order(dt$age)]
polygon(c( x,rev(x)), c(y2,rev(y1)), col = 'grey', border = NA)
lines(sort(dt$age), preds[ ,1][order(dt$age)], lty = 'solid', col = 'green')
lines(sort(dt$age), preds[ ,3][order(dt$age)], lty = 'dashed', col = 'red')
lines(sort(dt$age), preds[ ,2][order(dt$age)], lty = 'dashed', col = 'blue')
legend("bottom",c("fitted","upper bound","lower bound"),
       col=c("green","red","blue"), lwd=3)
```

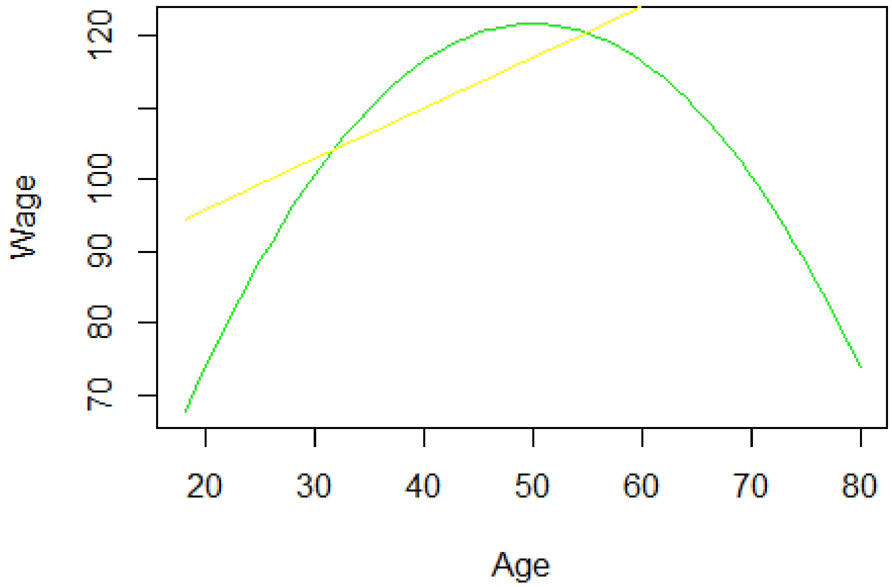


c.
The graphing output is a parabola shape. Age first positively affects wage but then starts to reduce wage at the age of 50. From the graph we can see that the 95% confident interval is narrowing down from age 20 to approximately age 35 but gradually expanding afterward. This make sense because middle-age people usually have more stable wage outcome compared with the younger and the older group
By fitting a polynomial regression, we are asserting a non-linear relationship between our predictor and predictee.

d.

```
model2<-lm(dt$wage~dt$age,data=dt)
summary(model2)

##
## Call:
## lm(formula = dt$wage ~ dt$age, data = dt)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -100.265  -25.115   -6.063   16.601  205.748
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 81.70474    2.84624   28.71   <2e-16 ***
## dt$age       0.70728    0.06475   10.92   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 40.93 on 2998 degrees of freedom
## Multiple R-squared:  0.03827,    Adjusted R-squared:  0.03795
## F-statistic: 119.3 on 1 and 2998 DF,  p-value: < 2.2e-16
```

```
plot(sort(dt$age), fitted(model1)[order(dt$age)],xlab="Age",ylab="Wage", col=
"green", type='l')
lines(sort(dt$age), fitted(model2)[order(dt$age)],col="yellow")
```



The linear regression function is: $\widehat{wage}2$=81.70474+0.70728*age.
This linear regression model is statistically significant, but the goodness of fit is even worse than the polynomial model. Age linear term is still statistically signifcant at 99% confident interval, but now age as constant positive effect on wage. More specifically, one unit increase in age will lead to increase in wage by 0.70728 unit.

Generally speaking, linear regression only captures the linear relationship between variables. Adding polynomial terms bring us more flexible to fit the data and therefore statistically more significant result. However, one drawback of polynomial is its complexity, as more higher order terms added to the model, the harder it becomes to interpret the relationship between Xs and Y.