Intro to Machine learning

Problem Set 1

Jingming Wei(Mark)

1/17/2020

Q1 Statistical and Machine Learning

Supervised learnings and unsupervised learnings are two main categories in machine learning, where each of them has its own purposes and practices in the real-world problem.

In supervised learning, we always know exactly what our dependent variable Y and the predictors X's are. We have two targets in doing supervised learning: prediction and inference. The first is accurately predicting Y using X's given, and the second is to gain a better understanding of the relationship between Y and X's. There is a true data generating process by which Y is generated by X's. There is a hypothetical data generating distribution according to which the data is distributed. We collect the data following some sampling procedure, and the data has an empirical distribution. Here, we impose a structural model between Y and X's that best fit the empirical distribution, and hopefully, the model can represent the true data generating process. The model can be either parametric or nonparametric. Parametric models include OLS, logistic regression, polynomial regressions, etc., and nonparametric models include KNN, Decision Trees, etc. The dependent variable can be continuously numeric or categorical, where the previous is generally called the regression problem, and the latter is called the classification problem. In supervised learning, the learning conceptually means selecting models with features by some optimization algorithm and then evaluating the model's performance post fitting. Specifically, we attain estimates of the model parameters using the train set. We then use the estimates to get predicted Y values in the test set to compare with the actual Y values. Take linear regression as an example. Suppose there is a linear relationship between Y and X1, X2, . . . , Xn, and our goal is to predict Y conditional on X's accurately. We can try different combinations of predictors and get estimates of the coefficients in each specification using the training set and compare the predicted values with the true values in the test set to decide the best combination of predictors which produce the best prediction results by some criterion. In this case, the process of getting the best combination of predictors and the corresponding coefficients is the learning process.

In unsupervised learning, in contrast, we don't know the response Y associated with a vector of X's. Our target, in this case, is more subjective. The goal is not always as simple as accurate predictions as a response. Instead, we try to find underlying relationships between X's or observations to identify interesting characteristics. For example, we might want to find possible groups to which each observation can be assigned. This type of task is called cluster analysis. Another example is the density estimation. While the supervised learning intends to infer a conditional distribution of X given Y, the unsupervised learning here tries to infer a priori probability distribution of X. Furthermore, it's more challenging to assess the unsupervised learning method, since we don't have a response Y as in supervised learning to compare the results. In other words, we don't know the true answers in unsupervised learning.

Q2 Linear Regression

We first attach the dataset convenient for later use and display the column names

```
> attach(mtcars)
> names(mtcars)
 [1] "mpg"  "cyl"  "disp" "hp"   "drat" "wt"   "qsec" "vs"   "am"   "gear"
[11] "carb"
```

a. The regression output shows that the coefficient associated with “cyl” is -2.8758, and the constant term is 37.8846.

```
> #fit the linear model using only "cyl" as a predictor
> lm.fit1 <- lm(mpg~cyl)
> summary(lm.fit1)

Call:
lm(formula = mpg ~ cyl)

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 predicted values of “mpg” are

```
> #get predicted values using the parameter estimated above
> pred.mpg <- predict(lm.fit1)
> pred.mpg
       1        2        3        4        5        6        7        8 
20.62984 20.62984 26.38142 20.62984 14.87826 20.62984 14.87826 26.38142 
       9       10       11       12       13       14       15       16 
26.38142 20.62984 20.62984 14.87826 14.87826 14.87826 14.87826 14.87826 
      17       18       19       20       21       22       23       24 
14.87826 26.38142 26.38142 26.38142 26.38142 14.87826 14.87826 14.87826 
      25       26       27       28       29       30       31       32 
14.87826 26.38142 26.38142 26.38142 14.87826 20.62984 14.87826 26.38142 
```

b. The statistical form of the regression model in the previous question is

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

c. The regression output shows that the coefficient associated with “cyl” is -1.5078, the coefficient associated with “wt” is -3.1910, and the constant term is 39.6863. We can see that the marginal effect of "cyl" on "mpg" decreases in absolute value by adding a new predictor. And also, a decrease in critical value of the coefficient of "cyl" suggests a lower significance. The R^2 increases, suggeting a better fit of the model.

```
> #fit the linear model using "cyl" and "wt" as predictors
> lm.fit2 <- lm(mpg~cyl+wt)
> summary(lm.fit2)

Call:
lm(formula = mpg ~ cyl + wt)

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. The regression output shows that the coefficient of "cyl" is -3.8032, the coefficient of "wt" is -8.6556, the coefficient of the interaction term "cyl·wt" is 0.8084, and the constant term is 54.3068. Both coefficients of "cyl" and "wt" are still negative and statitically significant, while the coefficient of "cyl" becomes more siginicant and "wt" becomes less significant after adding an interaction term. R^2 slightly increases. To include a multiplicative interaction term, we are actually asserting that a one-unit change in "cyl"(or "wt") will also change the effect of "wt"(or "cyl") on "mpg". In comparison, in the usual additive form, we are assuming that the effect of "cyl"(or "wt") on "mpg" is constant, which is independent of the value of "wt"(or "cyl").

```
> #fit the linear model with an interation term
> cylwt <- cyl * wt
> lm.fit3 <- lm(mpg~cyl+wt+cylwt)
> summary(lm.fit3)

Call:
lm(formula = mpg ~ cyl + wt + cylwt)

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 ***
```

```
cylwt         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
```

Q3 Non-linear Regression

We first import the "wage_data" and attach it for easy access.

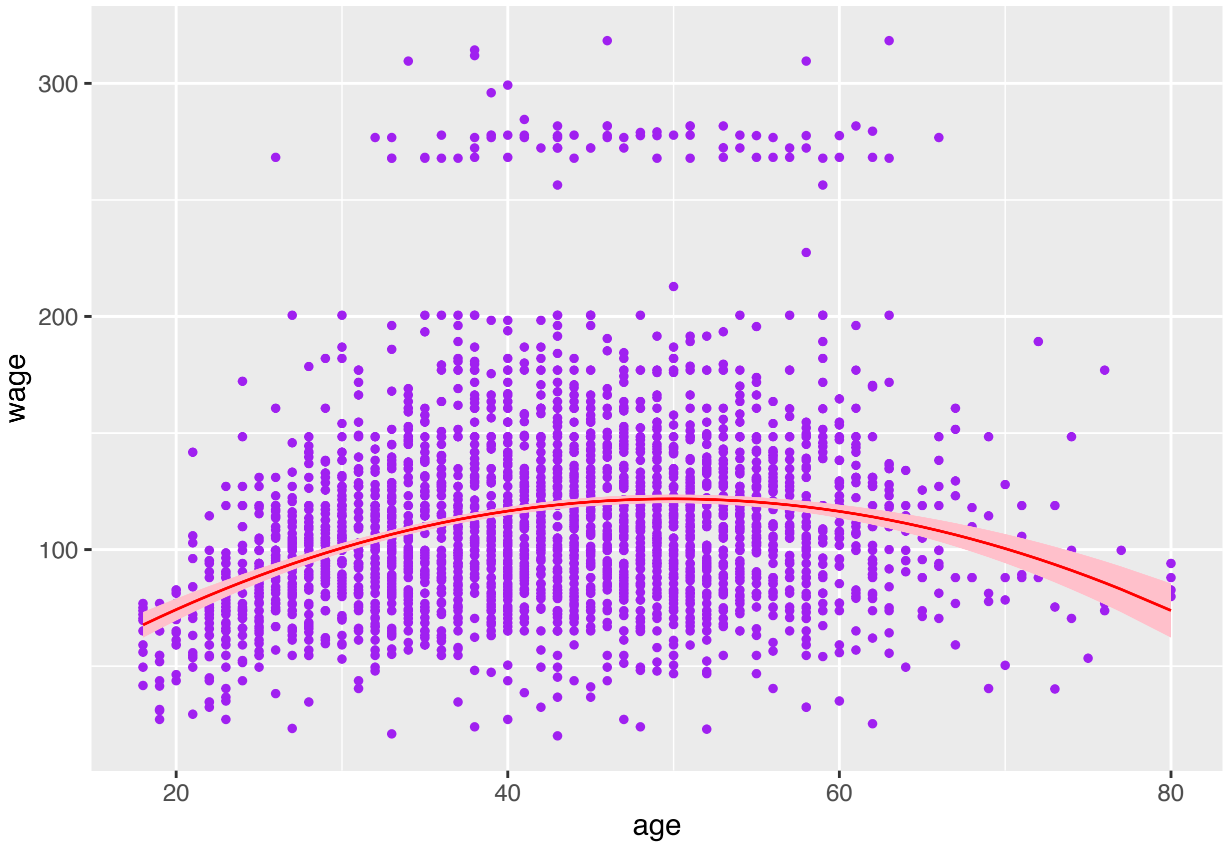
```
> wage_data <- read.csv("/Users/Mark/Desktop/Intro to Machine Learning/wage_data.csv")
> attach(wage_data)
> names(wage_data)
##  [1] "X"          "year"       "age"        "maritl"     "race"
##  [6] "education"  "region"     "jobclass"   "health"     "health_ins"
## [11] "logwage"    "wage"
```

a. The regression result shows that the coefficient of "age" is 5.294030, the coefficient of "age^2" is -0.053005, and the constant term is -10.425224. And both the two coefficients of "age" and "age^2" are statistically siginificant. Then we can say that, in general, age is positively associated with wage but the sign flips when we take the square of age. This suggests that there might be a non-linear relationship betweem wage and age.

```
> #fit the model using age and squared age as predictors
> square.fit <- lm(wage~age+I(age^2))
> summary(square.fit)
##
## Call:
## lm(formula = wage ~ age + I(age^2))
##
## 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
```

b. Please see below the output graph.

```
> library(ggplot2, warn.conflicts = FALSE)
> pred.wage <- data.frame(predict(square.fit, level=0.95, interval="confidence"))
> ggplot(wage_data,aes(age)) +
+   geom_point(aes(y=wage), color='purple', size=1) +
+   geom_ribbon(aes(ymin = pred.wage$lwr, ymax = pred.wage$upr), fill='pink') +
+   geom_line(aes(y=pred.wage$fit), color='red')
```



c. As we can see from the above graph, the predicted wage first increases until it reaches the maximum point where age is around 50. After the max, the predicted wage starts to decrease. The confidence interval band is pretty thin along the line, suggesting that our estimates have fairly small standard errors. But note that the points are pretty scattered at each value of age, meaning that there are lots of variations in wage even at a fixed level of age. Also, noticebly, the confidence interval bounds are larger at two tails. This is probably due to sparser data points at two tails or suggesting more variations in wage at two ends. By fitting a polynomial regression, we are asserting that there is a quadratic relationshiop between wage and age.

d. Here, we run a linear regression to compare with the results we get from part a.

```
> linear.fit <- lm(wage~age)
> summary(linear.fit)
##
## Call:
## lm(formula = wage ~ age)
##
## 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 ***
## 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
```

Statistically, we can see that the coefficients of "age" in both specifications are significant and positive. However, R^2 is larger in the quadratic regression than that in the linear one, suggeting that including a second order term improves the fit of the data. Substantively, a linear regression is not consistent with the true relationship between wage and age. People at early ages are more likely to see wage increase due to increased proficiency or job-hopping for better match. People in general become less productive after they reach a certain age level. Further, their wages should substantially decline after they retire. Therefore, a quadratic form might better reflect the true relationship between the two variables.