Problem Set 1 by Runhua Li

Runhua LI

2020/1/17

Statistical and Machine Learning

Both supervised learning and unsupervised learning are concepts related to "learning" from data, i.e., to extract information contained in data. That said, there are some major differences between supervised and unsupervised learning.

First, their goals are different. The goal of supervised learning is to predict potential outcomes using existing inputs. For example, to predict NBA match results using team stats and play stats. In this case, the measure of outcomes is explicitly given as WIN or LOSE, and the measure of inputs is also given. This case is a classification, meaning the outcome is a chatagorical variable, while the other type of supervised learning, i.e., regression, shares the same goal. However, the goal of unsupervised learning is to detect patterns associated with given inputs. For example, given a million pictures of cats and dogs, to recognize pictures of different objects, is an unsupervised learning task. Unlike supervised learning, there is no given measure of outcomes, because we don't label some inputs with "Cat" or "Dog" and then train the model. Instead, we let the model to discover by itself the patterns in those inputs. Therefore, the goals of supervised and unsupervised learning are different, as the former aims to predict, while the latter aims to recognize patters.

Second, the Xs and Ys and their relations in supervised and unsupervised learning are different. In supervised learning, there is a predictive relationship between X and Y, meaning we want to predict Y using X. By that, we are assuming a correlation, or even causal relationship, between Y and X with preexisting measures, otherwise we wouldn't expect X to be a predictor of Y. In unsupervised learning, take cluster analysis as an example, we don't even know in advance what outcomes would be generated from the learning process. Outcomes in unsupervised learning is rather a result of the learning process, which means we don't assume explicit and meaningful relationship between outcomes and inputs.

Third, evaluations, i.e., the criteria for "good learning", of supervised and unsupervised learning can be different. As covered in the lecture, we can compare trained supervised learning models with testing dataset and calculate their MSEs (mean squared error for regressions) or confusion matrices (for classifications). This, however, is not necessarily applicable to evaluate unsupervised learning. This is because there is not preexisting explicit measure of outcomes in unsupervised learning. As a result, the performance of unsupervised learning unavoidably involve subjective judgement.

In all, it is important to note the differences between supervised and unsupervised learning. In choosing which kind of learning to use, it is always necessary to be clear of our goal. With that in mind, we should also be mindful of their difference in the relations between Xs and Ys, as well as their evaluations. When generating data for our project, we should be guided by our understanding of them.

Linear Regression

a. Predicting Miles-per-gallon as A Linear Function of Cylinders

```
lmod <- lm(mpg ~ cyl, data = mtcars)
summary(lmod)
```

```
##
## 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
```

```
lmod.pred <-  predict.lm(lmod, mtcars)
lmod.pred
```

```
##           Mazda RX4       Mazda RX4 Wag          Datsun 710
##            20.62984            20.62984            26.38142
##      Hornet 4 Drive   Hornet Sportabout             Valiant
##            20.62984            14.87826            20.62984
##          Duster 360           Merc 240D            Merc 230
##            14.87826            26.38142            26.38142
##            Merc 280           Merc 280C          Merc 450SE
##            20.62984            20.62984            14.87826
##          Merc 450SL         Merc 450SLC  Cadillac Fleetwood
##            14.87826            14.87826            14.87826
## Lincoln Continental   Chrysler Imperial            Fiat 128
##            14.87826            14.87826            26.38142
##         Honda Civic      Toyota Corolla       Toyota Corona
##            26.38142            26.38142            26.38142
##    Dodge Challenger         AMC Javelin          Camaro Z28
##            14.87826            14.87826            14.87826
##    Pontiac Firebird           Fiat X1-9       Porsche 914-2
##            14.87826            26.38142            26.38142
##        Lotus Europa      Ford Pantera L        Ferrari Dino
##            26.38142            14.87826            20.62984
##       Maserati Bora          Volvo 142E
##            14.87826            26.38142
```

As is shown in the summary of the linear model,the "intercept" is 37.8846, whereas the coefficient of predictor *cyl* is -2.8758. Predicted values lmod.pred are fitted outputs of the model.

b. Statistical Form

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

Where i indexes observations. ϵ is the error term whose expected value is 0 conditional on cyl.

c. Adding Weight into the Model

```
lmod2 <- lm(mpg ~ cyl + wt, data = mtcars)
summary(lmod2)
```

```
##
## 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
```

```
lmod2.pred <- predict.lm(lmod2, mtcars)
```

Compared to the previous linear model, the coefficient of predictor cyl is less negative, i.e., with less absolute value. This can be an indicator that the previous specification contains omitted-variable bias. The "statistical significance" of the coefficient of cyl also decreased.

The coefficient of the new predictor wt is statistically significant, and significant in its absolute value. The R^2 of this model is also higher than the previous one, implying this can be a better fitting model.

d. Adding the Interaction Term

```
lmod3 <- lm(mpg ~ cyl + wt + cyl * wt, mtcars)
summary(lmod3)
```

```
##
## 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
```

```
lmod3.pred <- predict.lm(lmod3, mtcars)
```

Both coefficients of predictor *cyl* and *wt* remain negative, meaning the more cylinders or weight of the car, the less miles-per-gallon predicted by our models. However, the magnitudes of both coefficients increased, which is due to the addition of the interaction term.

By including the interaction term in the function, we are asserting that the effect (relation) of predictor *cyl* on outcome *mpg* is dependent on the weight of the car. Similarly, we are also asserting that the effect of predictor *wt* on *mpg* is dependent on the number of cylinders the car have.

Non-linear Regression

a. Fitting the Polynomial Model

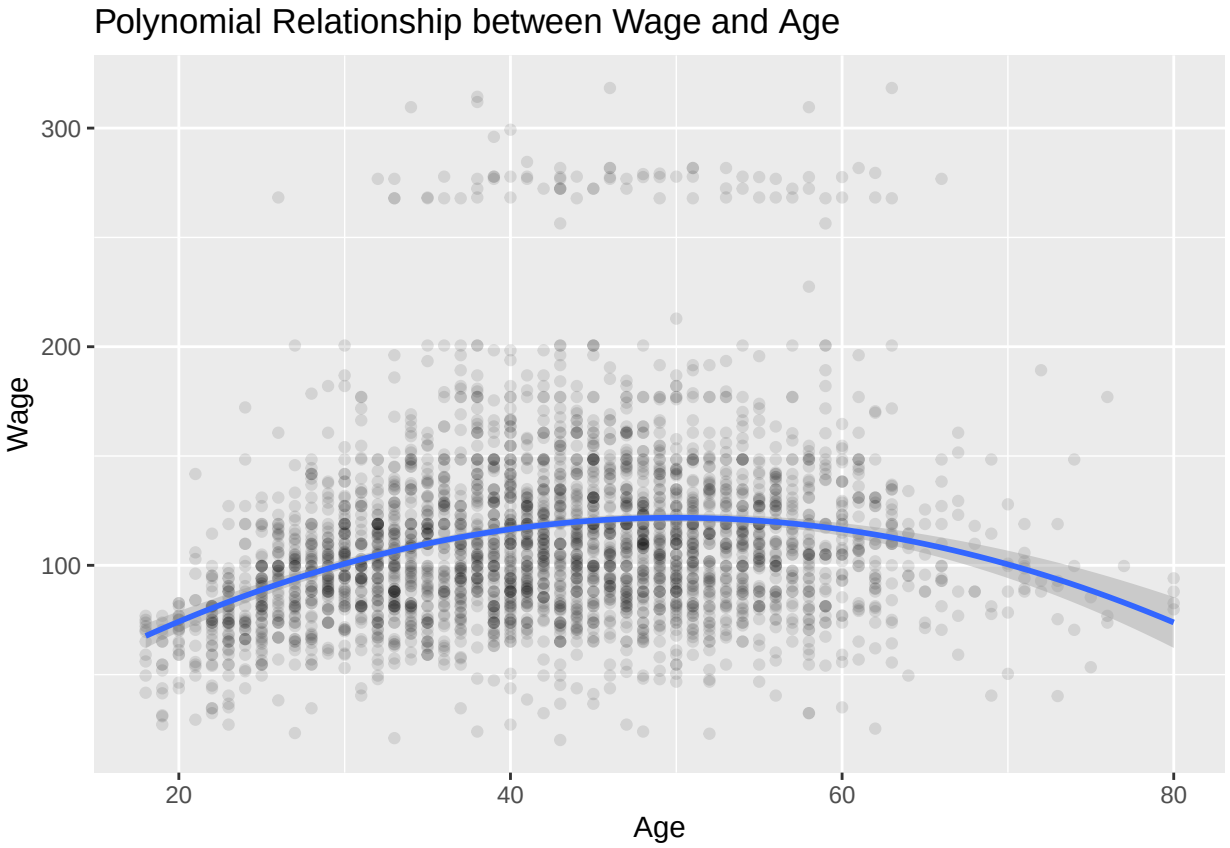
```
polymod <- lm(wage ~ poly(age, 2, raw = TRUE), wage.data)
summary(polymod)
```

```
##
## Call:
## lm(formula = wage ~ poly(age, 2, raw = TRUE), data = wage.data)
##
## 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, 2, raw = TRUE)1   5.294030   0.388689  13.620   <2e-16 ***
## poly(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
```

As is shown in the summary, though this age polynomial model is not capturing very much of the variance of wage (the adjusted R^2 is less than 0.1), both the coefficients of age and age^2 are statistically significant. The result demonstrates the predictive relationship between wage and age as a downward-opening quadratic function.

b. Plotting the Confidence Interval

```
ggplot(data = wage.data, aes(x = age, y = wage)) +
  geom_point(alpha=0.1) +
  geom_smooth(formula = y ~ poly(x,2),
              method = lm,
              level = 0.95) +
  labs(x = "Age",
       y = "Wage",
       title = "Polynomial Relationship between Wage and Age")
```



c. Discussing the Output

The graph tells that we predict wage to be increasing with respect to age before around age 50, after which wage decreases with age.

By using this second order polynomial model, we are asserting that the relationship between wage and age can be approximated by a quadratic function.

d. Discussing the Difference

Statistically, a polynomial regression model can seem similar to a linear regression model in its function form. However, it is different because, unlike linear models in which predictors are assumed to be independent from each other (though not necesarrily super realistic), a single feature's polynomials with different degrees each has its own coefficient.

This also makes the interpretation of coefficients more subtle in polynomial regressions, as we cannot interpret the coefficient of *age* as "the predictive effect of *age* on *wage* holding other independent variables constant". Instead, one should evaluate the effect of one feature on the outcome based on all coefficients of its polynomials.

Practically, both model can be useful under different settings. It depends on our understanding and assumption of the underlying data generating process to make an initial choice of which model to use. Linear regression models are based on more restrictive assumptions, and are therefore potentially biased. On the other hand, though polynomial regression models can be more flexible, there is also the risk of over-fitting the training data, especially when the order of the polynomial model becomes large.

To finally decide which model to use for prediction, testing the two kinds of model in test data with resepct to MSE can be a good approach.