Problem Set 1

Soowon Jo

1/15/2020

##Problem Set 1: Learning and Regression

Statistical and Machine Learning

####1. Describe in 500-800 words the difference between supervised and unsupervised learning.

There are two different types of learning tasks in the field of machine learning: supervised and unsupervised. Supervised learning is performed using ground truth, meaning that we have prior knowledge of what the desired output values should be. Thus it is a process of learning when a model is trained on a labelled dataset in which the inputs are paired with the correct outputs. After training, a supervised learning algorithm will take in new unseen inputs and will determine which label the new inputs will be classified as based on prior training data.

There are two subcategories in supervised learning: classification and regression. The unique difference between the two is the shape of the response feature. Classification helps us to map input values to output labels by assigning them a discrete class or category. Classification includes binary classification in which the task is to classify the input values into two groups (predicting which group each value belongs to) and multiclass classification in which the task is to classify the input values into more than two classes; e.g., classify a set of images of animals which may be zebra, cat, lion, or bear.

A regression, on the other hand, is a predictive statistical process where the model attempts to predict a continuous dependent variable from a number of independent variables. The main goal for both classification and regression, therefore, is to approximate the mapping function that allows us to effectively predict the output variables by using new input data.

Unsupervised learning does not have labelled outputs so that we need to make an inference of the natural structure present within a set of data points. The main goal of this task is to group unsorted information according to similarities, patterns, and differences without any prior training of data. The two most common categories of algorithms used in unsupervised learning are clustering and association rule mining. Clustering is the process of grouping similar entities together. It helps us to reduce the dimensionality of the data when dealing with a copious number of variables. The two most common algorithms used in clustering include K-means clustering and hierarchical clustering. The objective of K-means is to find the number of centroids that represents the center of the clusters in a dataset. The algorithm identifies k number of centroids and allocates every data point to the nearest cluster while keeping the centroids as small as possible. Hierarchical clustering, on the other hand, begins by assigning each data point as a separate cluster. Then, it identifies and merges the two clusters that are closest together and repeat the same process until one cluster or K clusters are formed.

Association rule is another important unsupervised learning method used in machine learning. Unlike clustering, it identifies relationships between a set of elements present in different entries in datasets. It works based on antecedent (if) and consequent (then) statements, which help to reveal associations between the independent variable in a dataset, relational dataset or other information repositories. An antecedent is something to be found in data while a consequent is something that is found in combination with the antecedent. For instance: “If a customer purchases a diaper, she’s 80% likely of buying a beer.” Here, the

diaper is the antecedent and beer is the consequent. Association rules, thus, are created by thoroughly analyzing data and looking for frequent if/then patterns and help us understand how or why such products or items are frequently purchased together.

Linear Regression Regression

####1. Using the mtcars dataset in R (e.g., run names(mtcars)), answer the following questions:

1a. Predict miles per gallon (mpg) as a function of cylinders (cyl). What is the output and parameter values for your model?

```
library(tidyverse)
```

```
##   Attaching packages                                           tidyverse 1.3.0

##   ggplot2 3.2.1      purrr   0.3.3
##   tibble  2.1.3      dplyr   0.8.3
##   tidyr   1.0.0      stringr 1.4.0
##   readr   1.3.1      forcats 0.4.0

##   Conflicts                                                tidyverse_conflicts()
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
mtcars <- mtcars
# build linear regressin model on data mtcars
mpg_cyl_regress <- lm(mpg ~ cyl, data = mtcars)
summary(mpg_cyl_regress)
```

```
##
## 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
```

This model has two coefficients, the intercept and the main effect for cylinder.

The intercept of 37.88 is the miles per gallon a car is expected to have when we consider the average cylinder of all cars in the dataset.

The coefficient for cylinder is the slope of regression line (in other words, the effect cylinder has in mpg). The slope of -2.87 means that for every 1 cylinder decreases roughly by 3 miles per gallon.

The output informs us there is a relationship between mpg and cylinder.

1b. Write the statistical form of the simple model in the previous question (i.e., what is the population regression function?).

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

1c. Add vehicle weight (wt) to the specification. Report the results and talk about differences in coefficient size, effects, etc.

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

```
##
## 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
```

The coefficient for cylinder is -1.50 whereas that for weight variable is -3.19. Both correlation coefficients are negative meaning that they both shows an inverse relationship between the two parameters (mpg and either number of cylinders or weight) tested. According to the result, the average effect of weight is bigger than that of cylinder. The miles per gallon(mpg) decreases by 1.5 for every unit change in the number of cylinders while mpg decreases by 3.19 for every unit of weight of all vehicles.

Here, the Pr(>|t|) or p-values for cylinder and weight are 0.001 and 0.0002, respectively, which are small enough (< 0.05) to consider both coefficients significant so that this model is indeed statistically significant. Given that p-value of weight is less than 0.001 whereas that of cylinder is less than 0.01, p-value of weight provides stronger evidence against the null hypothesis.

Note: Compared to the model that only has cylinder as an independent variable and its coefficient for cylinder is -2.87, the second model that includes weight variable reduced the effect size for the coefficient for cylinder to -1.5 that is closer to zero. This difference tells us that the model accounts weight more than cylinder variable to explain the variance in the dependent variable.

1d. Interact weight and cylinders and report the results. What is the same or different? What are we theoretically asserting by including a multiplicative interaction term in the function?

```r
mpg_cyl_wt_regress2 <- lm(mpg ~ cyl*wt, data = mtcars)
summary(mpg_cyl_wt_regress2)
```

```
##
## Call:
## lm(formula = mpg ~ 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
```

The magnitude of coefficients for all intercept, cylinder, and weight have increased. The second coefficient is the slope between mpg and the number of cylinder when the average weight of all vehicles is equal to 0 while the third coefficient is the slope between mpg and weight when the number of cylinder is equal to 0. The last coefficient is the change in the slope as one of the two variables increases. For instance, if the number of cylinder increases by one the slope between mpg and weight increases by 0.8. In other words, the effect of weight on mpg increases by 0.8 under increasing number of cylinders compared to the effect of weight under control cylinder condition.

The p-values in the output tell us that the interaction effect (cyl*wt) is statistically significant. Consequently, the mpg we derive from weight depends on the number of cylinders. Also, note that the R-squared adjusted value increased from the earlier model, indicating a better fit.

Non-linear Regression

####1. Using the wage_data file, answer the following questions:

1a. Fit a polynomial regression, predicting wage as a function of a second order polynomial for age. Report the results and discuss the output (hint: there are many ways to fit polynomials in R, e.g., I, ^, poly(), etc.).

```r
wage_data = read.csv("/Users/soowonjo/Desktop/MachineLearning/PB1/wage_data.csv")
wd_model <- lm(wage ~ age + I(age^2), wage_data)
summary(wd_model)
```

```
##
## Call:
## lm(formula = wage ~ age + I(age^2), data = wage_data)
##
## Residuals:
```

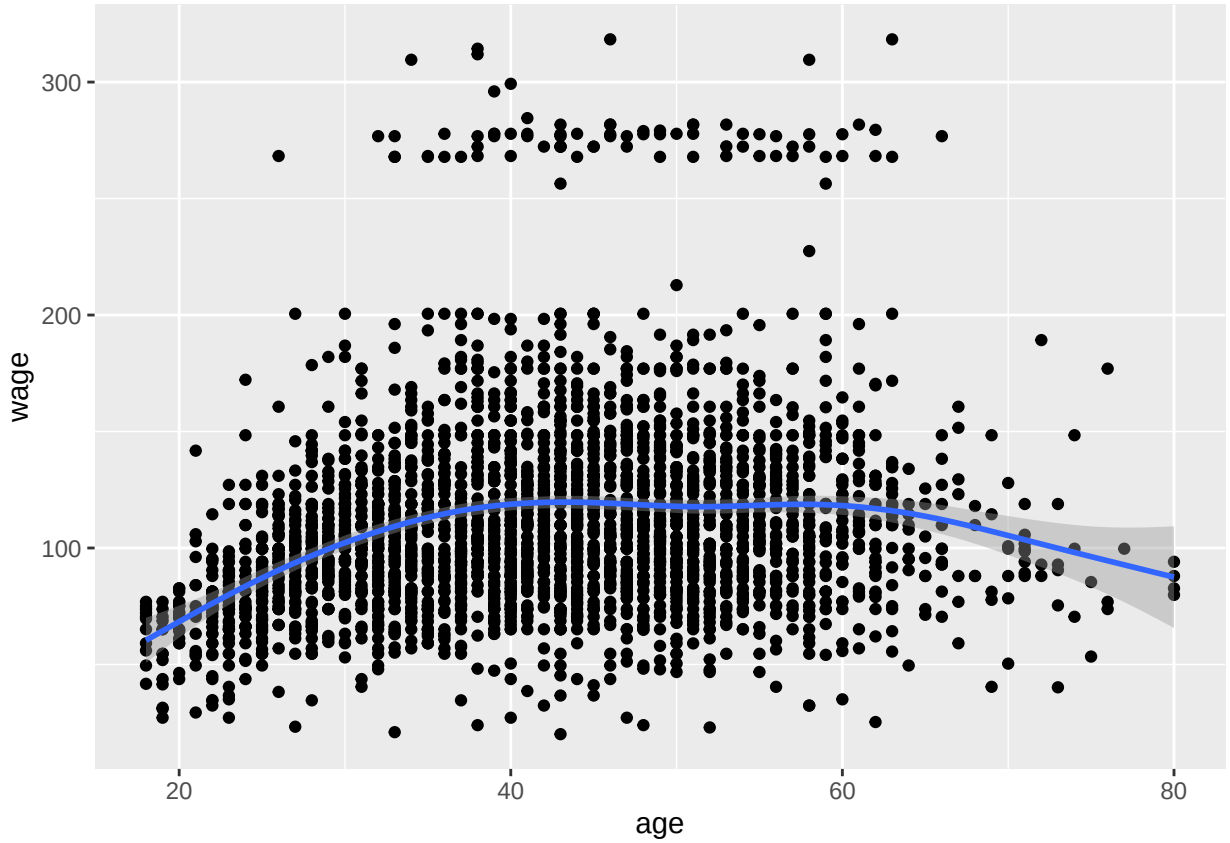
```
##     Min      1Q  Median      3Q     Max
## -99.126 -24.309  -5.017  15.494 205.621
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -10.425224   8.189780  -1.273    0.203
## age           5.294030   0.388689  13.620   <2e-16 ***
## I(age^2)     -0.053005   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
```

The summary output states that coefficients for both the second and the third term are statistically significant. This implies that both models fit the data significantly well. The R-squared indicate that 8.2% of the variability of the data is explained by a polynomial of degree 2, instead of 8.1% with a polynomial degree of 1. The standard deviation of the residual error is 39.99 (wage) with this model.

1b. Plot the function with 95% confidence interval bounds.

```
library(ggplot2)
wage_plot <- wage_data %>% ggplot(aes(age,wage))+
  geom_point()+
  stat_smooth(formula = wd_model$formula, level = .95)
wage_plot
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



1c. Describe the output. What do you see substantively? What are we asserting by fitting a polynomial regression?

According to the output above, people who are in the age range between 40~50 have a higher wage. Using the linear regression may not have sufficiently captured the pattern or relationship between the two variables (wage and age), causing the under-fitting problem. To overcome this issue, I increased the complexity of the model by fitting a polynomial regression. Polynomial regression extends the linear model by adding extra predictors, obtained by raising each of the original predictors to a power. The wd_model I used here could still be considered as a linear model, but the curve I am fitting is quadratic in nature. As a result, the curve can fit the data better than the linear line. The polynomial regression models thus can be used to approximate a complex nonlinear relationship and allow us to make assumptions about the shape of data.

1d. How does a polynomial regression differ both statistically and substantively from a linear regression (feel free to also generalize to discuss broad differences between non-linear and linear regression)?

Both types of regression models find lines or curves that fit the data. Linear regression model is linear in the parameters which have to be estimated. It attempts to model the relationship between variables by fitting a linear equation to observed data. Non-linear regression, on the other hand, generates a line (typically a curve) as if every value of Y was a random variable. The goal of the model is to make the sum of the squares as small as possible. The sum of squares is a measure that tracks how much observations vary from the mean of the dataset. Nonlinear regression models is more complicated than linear regression models to develop because the function is created through a series of approximations that stem from trial-and-error. Also, non-linear regression models are more prone to commiting overfitting which may cause the regression coefficients, p-values, and R-squared to be misleading.