Problem Set 1

Minyoung Do

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.

Both supervised and unsupervised machine learnings have three main components: data, features, and algorithms. In supervised learning, a human "supervisor" has to label and wrangle data as well as to adjust the algorithms and find the best fitting model. On contrary, unsupervised learning is literally leaving machines unsupervised; the machine learns on their own. While supervised learning has both input and output data, unsupervised learning only has the input data. It is rather clear what relationships you want to improve from the model in supervised learning, as it already has independent and dependent variables. However, in unsupervised learning, the goal is to have the machine find out how the relationships or correlations look like. Unsupervised model suggests ways to improve how to classify/label the data as well, whereas supervisors have to come up with a way to improve the model in supervised learning. Additionally, in the case of unsupervised learning, it is unpredictable at times, as it does not involve any supervision. For example, while a classification algorithm in supervised learning simply assigns labels fed by the supervisor to data, the unsupervised algorithm will categorize the data into groups in accordance with the observed similarities among the data, assigning its own label to the groups. Therefore, in the supervised learning case, the supervisor should alter the labels as the data changes, while unsupervised learning does not need modification as it works on its own. Supervised learning is very commonly used as the machine learns quicker than unsupervised.

The model is trained on the training set in supervised learning; in other words, machines learn from the training set as an example to make predictions about unknown data. It uses an existing data set to train a model to predict the relationship outside the data we already know. Thus, the

machine uses a part of the data set to learn, and the other part of data to validate and measure how accurate the model is, which is called test set. Supervised learning requires knowledge in the algorithm's possible outputs and the correctly labelled data to train on. Supervisors need to constantly make adjustments and rebuild the model to make sure it is optimal and improves its predictive power. However, unsupervised learning does not use human guidance along the process; it learns to detect patterns and relationships that are not easily identifiable by humans. This, therefore, means that unsupervised learning often solves problems that are too complicated for humans. One of the most famous examples of unsupervised learning is Facebook's tag feature. It would have been impossible to manually identify who are in the photos and tag them; but using unsupervised algorithm, Facebook adopted a recommendation system that automatically guesses who it is based on analyzing and grouping images. It learns from all the images people in your friends list have posted, analyze faces and postures to train itself to identify individuals if they're present in any picture. In contrast, supervised algorithm would learn information, such as facial features and posture, from the image data it already has, and try to recognize individuals in the new photos.

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?

```
# running a simple linear regression
first <- lm(mpg ~ cyl, data = mtcars) %>%
  tidy()
# turning the regression result into a table for presentation
kable(first) %>%
  kable_styling("striped", full_width = F, position = "left") %>%
  column_spec(1, bold = T, background = "pink") %>%
  add_footnote("DV: Miles Per Gallon (mpg), IV: Cylinders (cyl)",
```

```
                notation = "alphabet")
```

term	estimate	std.error	statistic	p.value
(Intercept)	37.88458	2.0738436	18.267808	0
cyl	-2.87579	0.3224089	-8.919699	0

[a] DV: Miles Per Gallon (mpg), IV: Cylinders (cyl)

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

Y_i = 37.885 - 2.876X_i

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

```
# running a multiple regression
second <- lm(mpg ~ cyl + wt, data = mtcars) %>%
  tidy()
# turning the regression result into a table for presentation
kable(second) %>%
  kable_styling("striped", full_width = F, position = "left") %>%
  column_spec(1, bold = T, background = "pink") %>%
  add_footnote("DV: Miles Per Gallon (mpg), IV: Cylinders (cyl), Vehicle Weight (wt)",
               notation = "alphabet")
```

term	estimate	std.error	statistic	p.value
(Intercept)	39.686262	1.7149840	23.140893	0.0000000
cyl	-1.507795	0.4146883	-3.635972	0.0010643
wt	-3.190972	0.7569065	-4.215808	0.0002220

[a] DV: Miles Per Gallon (mpg), IV: Cylinders (cyl), Vehicle Weight (wt)

As reported in the table above, the coefficient for `cylinders` is -1.5, while it is -3.19 for `vehicle weight`. This indicates the effect of `cylinders` is smaller than the one of `vehicle weight` in the estimated model. The miles per gallon decrease by 1.5 for every unit change in the number of cylinders ($p < 0.001$). At a 5% significance level, the weight of vehicle also has a negative correlation;

with an increase in every unit of vehicle weight, the miles per gallon will decrease as well by 3.19.

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?

```
# interactions in regression
third <- lm(mpg ~ cyl + wt + cyl*wt, data = mtcars) %>%
  tidy()
# turning the regression result into a table for presentation
kable(third) %>%
  kable_styling("striped", full_width = F, position = "left") %>%
  column_spec(1, bold = T, background = "pink")
```

term	estimate	std.error	statistic	p.value
(Intercept)	54.3068062	6.127535	8.862749	0.0000000
cyl	-3.8032187	1.005028	-3.784193	0.0007472
wt	-8.6555590	2.320122	-3.730648	0.0008610
cyl:wt	0.8083947	0.327322	2.469723	0.0198824

What I infer from the coefficients is that the effect of `vehicle weight` is bigger than the one of `cylinders` in the model. Both variables have a negative correlation with the miles per gallon. The miles per gallon would decrease by 3.8 for every unit change in the number of cylinders, while an increase in one unit of vehicle weight would decrease the miles per gallon by 8.65.

By adding `cyl*wt` in the formula, this regression model attempts to measure the interaction effect between `cylinders` and `vehicle weight`. The coefficients are different from the result of multiple regression in **1c** because, in this case, the effect of one independentt variable changes depending on the value of the other independent variable. Thus, the coefficient for `cyl*wt` shows one variable's effect on the miles per gallon depending on the other variable.

Non-linear Regression

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

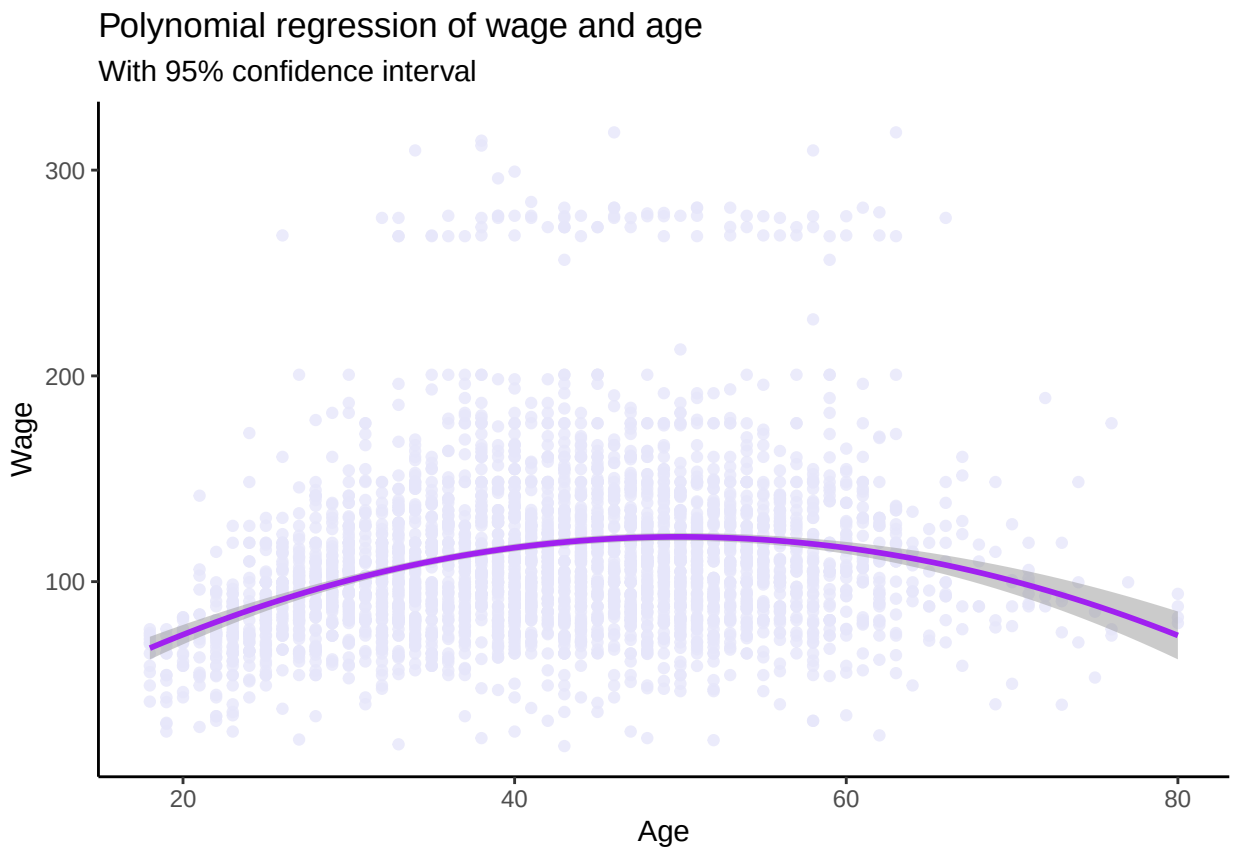
1a. Fit a polynomial regression, predicting wage as a function of a second order polynomial for age. Report the results and discuss the output (hint: there are many ways to fit polynomials in R, e.g., I, ^, poly(), etc.).

```
wage <- read_csv("s3.csv")
lm(wage ~ age + I(age^2), data = wage) %>%
  summary()
```

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

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

```
wage %>%
ggplot(aes(age, wage, group = 1)) + # what do you see?
  geom_point(alpha = .8, color = "lavender") +
  stat_smooth(method = lm, formula = y ~ poly(x, 2), color = "purple", alpha = 0.5) +
  labs(title = "Polynomial regression of wage and age",
       subtitle = "With 95% confidence interval",
       x = "Age",
       y = "Wage") +
  theme_bw() +
  theme(axis.line = element_line(color = "black"),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.border = element_blank(),
        panel.background = element_blank())
```



1c. Describe the output. What do you see substantively? What are we asserting

by fitting a polynomial regression?

The plot shows a concave regression line, which indicates the relationship between wage and age is linear but not best represented by a simple linear regression line; rather, the relationship is better fitted by a curvilinear line. The wage tends to increase towards middle age, until they hit approximately 50, and then decrease afterwards.

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)?

Polynomial Regression can fit a relationship between the independent variable and dependent variable, modeled as an n^{th} degree polynomial. Polynomial regression is a type of linear regression, but fits a non-linear relationship between the independent and dependent variables. Depending on the shape of the data, a polynomial regression could provide a better approximation than a linear one, and also be a better fit in order to minimize the error. It explores beyond linear relationships and fits more curved lines by including higher order powers of an independent variable.

While a linear regression model has one configuration, non-linear regression can fit a variety of different curves. The parameters should be linear in linear regression, but it is possible to fit a curve by rasiting an independent variable by an exponent, e.g. using a squared or cubed term. However, nonlinear regression provides a more flexible curve-fitting functionality than linear regression in terms of the shapes of the curves that it can fit. The downside is that it might take a few tries before choosing a right function that has the best fit for the relationship. Also, because polynomial regression is more flexible, it is more sensitive to outliers as well.