Homework 1

Michael Carrion

1/17/2020

Problem 1

Within the realm of machine learning, supervised and unsupervised learning account for approximately 90% and 8%, respectively, of all machine learning tasks. The main difference between the two methods stems from the type of data available. In supervised learning, we have access to data which is well-labeled. That is, we have prior knowledge of what the output value Y_i is for each sample X_i. Unsupervised models, as we will examine shortly, help discover patterns in an "unlabeled" dataset (that is, one where features have no associated response).

Given a training set consisting of features and corresponding outputs, supervised learning is the method of deriving a (hypothesis) function that best models the true relationship in the data (termed the population function). This technique is typically used in two contexts: classification and regression. In classification, we deal with qualitative (discrete) data, which take on values in one of K different classes. For example, a classification problem may seek to predict whether or not an individual contracts a specific disease (termed binary classification), or may predict the nationality of a person based on a variety of factors such as height, job, age, education, etc. (termed multiclass classification). While a variety of algorithms may be used for classification problems, some of the most common are logistic regression and naïve Bayes. Regression, on the other hand, deals with quantitative data. For example, we may use regression to predict the market price of a house, or the future price of a stock. Linear regression, ridge regression, and support vector machines are common regression algorithms.

With such a wide range of algorithms available for supervised learning tasks, it's important to take into account several factors when selecting a supervised learning algorithm. Firstly, it's important to consider what is known as the bias-variance tradeoff: An algorithm is said to be particularly biased if it is largely incorrect in predicting the correct output for a feature X_i; high bias can cause an algorithm to miss the relevant, underlying relationships between inputs and outputs. On the other hand, an algorithm is said to have high variance if it is very sensitive to changes in the training set (that is, it predicts largely different outcomes when trained on different training sets); high variance can cause the algorithm to model the random noise in the data, as opposed to the actual relationship between features and outputs, a concept known as overfitting. In general, a more flexible model has higher variance and lower bias. Second, it's important to consider the "Curse of Dimensionality" in selecting a model: In high dimensions (that is, inputs with a large number of features), even comparatively close neighbors of an observation tend to be far away. Thus, if the input feature vectors tend to have a very high dimension, the model will likely have an unusually high variance (even if the true population function only depends on a select few of the features). Several unsupervised methods (e.g. principal component analysis) can be used to reduce the dimension of the inputs, as we will soon see.

Unsupervised learning, on the other hand, is used when data is not well labeled. That is, for each predictor X_i there is no associated output Y_i. Instead, we seek to infer some patterns or structure from our data without using explicitly provided labels. However, it's important to recognize that since no output values are provided, it's more difficult to assess model accuracy; in supervised learning, however, we can use metrics to assess the proportion of our test set that was classified correctly, or we can examine the root mean squared error of our regression model to assess accuracy. However, unsupervised learning has several important applications, namely clustering and dimensionality reduction. Clustering, or "exploratory data analysis" helps discover previously unknown patterns in the data. For example, a marketing company trying to segment a population into smaller groups with similar demographics may use a clustering algorithm such as K-means clustering. Dimensionality reduction, on the other hand, reduces the number of features under consideration, removing the redundant or unimportant features. As this eliminating of redundant features is a key task in data processing, a dimensionality-reducing algorithm such as principal component analysis may first be applied to the data before feeding it into a supervised learning model.

Ultimately, both supervised and unsupervised learning methods have their strengths and weaknesses. The choice of which method to use depends on a variety of factors, such as the type of data available, the model complexity, the goal of the data analysis, etc., but both play critical roles in the field of machine learning.

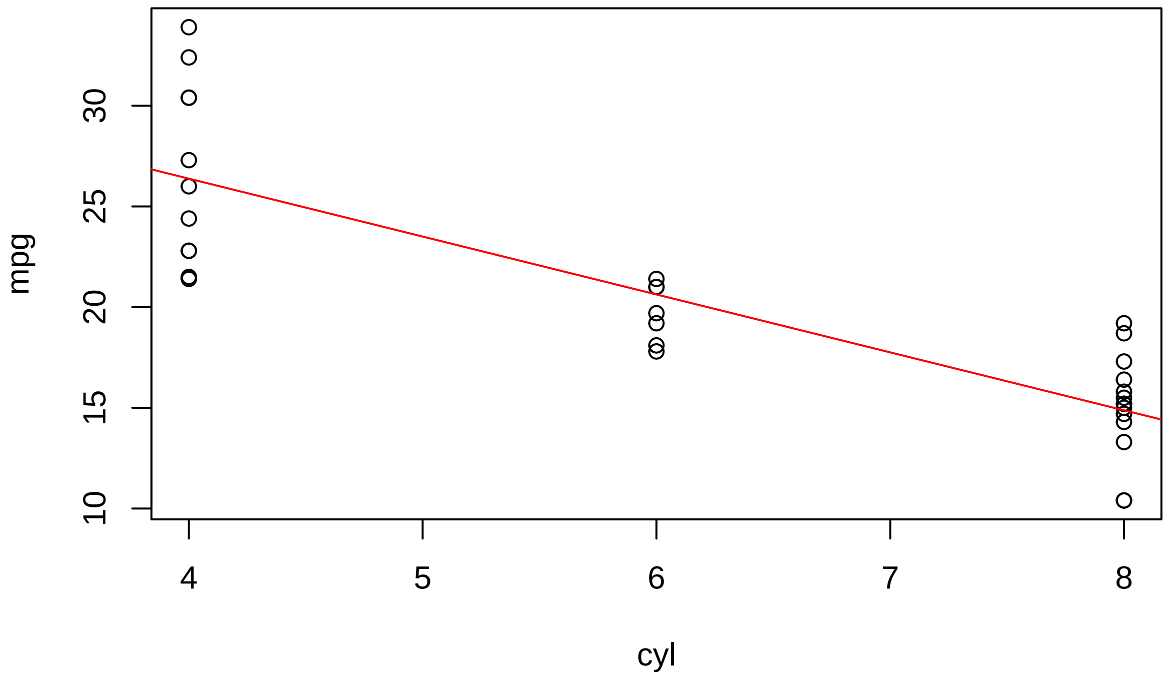
Problem 2

(a) A plot of the data (see below) suggests that a linear model does in fact approximate the data well.

```r
myCars <- mtcars
myOLS <- lm(mpg ~ cyl, myCars)
summary(myOLS)
```

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

```r
plot(mpg~cyl,myCars)
abline(myOLS,col="red")
```



We see that our parameter estimates are $\hat{\beta}_0 = 37.88$ and $\hat{\beta}_1 = -2.88$.

b) From above, we see our model is of the form

$$\hat{Y} = 37.88 - 2.88X$$

where Y denotes mpg and X denotes the number of cylinders of the vehicle. Thus, we see that an additional cylinder is usually associated with a drop of 2.88 mpg for the vehicle.

c) Adding vehicle weight, we get

```
myOLS2 <- lm(mpg ~ cyl + wt, myCars)
summary(myOLS2)
```

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

This suggests the following modified population regression function

$$\hat{Y} = 39.69 - 1.51X_1 - 3.19X_2$$

where X_1 denotes the number of cylinders and X_2 denotes the vehicle weight. Thus, we can see that, when we take vehicle weight into account, the effect of having extra cylinders plays less of a role in predicting mpg (though it is still negatively associated with mpg). Further, the results also suggest that vehicle weight as a larger negative effect on mpg (that is, the heavier vehicles have reduced mpg). Further, the adjusted r-squared value increased from 0.72 to 0.81, suggesting that this model accounts for more of the variability in the data, and is thus a better fit.

d) With the interaction term weight x cylinders, we have

```
myOLS3 <- lm(mpg ~ cyl + wt + (cyl*wt), myCars)
summary(myOLS3)
```

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

This suggests the following model, where X_1 denotes the number of cylinders, X_2 denotes the vehicle weight, and X_3 denotes the interaction term.

$$\hat{Y} = 54.30 - 3.80X_1 - 8.66X_2 + 0.81X_3$$

This model has a much larger intercept than the previous model, and greatly reduced coefficients on X_1 and X_2 (though the sign on each is the same). Further, the adjusted R-squared slightly increased, suggesting that this model may provide a better fit. The interaction term suggests that the relationship between weight and vehicle mpg is different based on the amount of cylinders a car has.

Problem 3

(a)

```
wages <- read.csv("/Users/Michael/Downloads/wage_data.csv")
age2 <- wages$age^2
myOLS4 <- lm(wage ~ age + age2, wages)
summary(myOLS4)
```

```
##
## Call:
## lm(formula = wage ~ age + age2, data = wages)
```

```
##
## 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 ***
## age2         -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
```

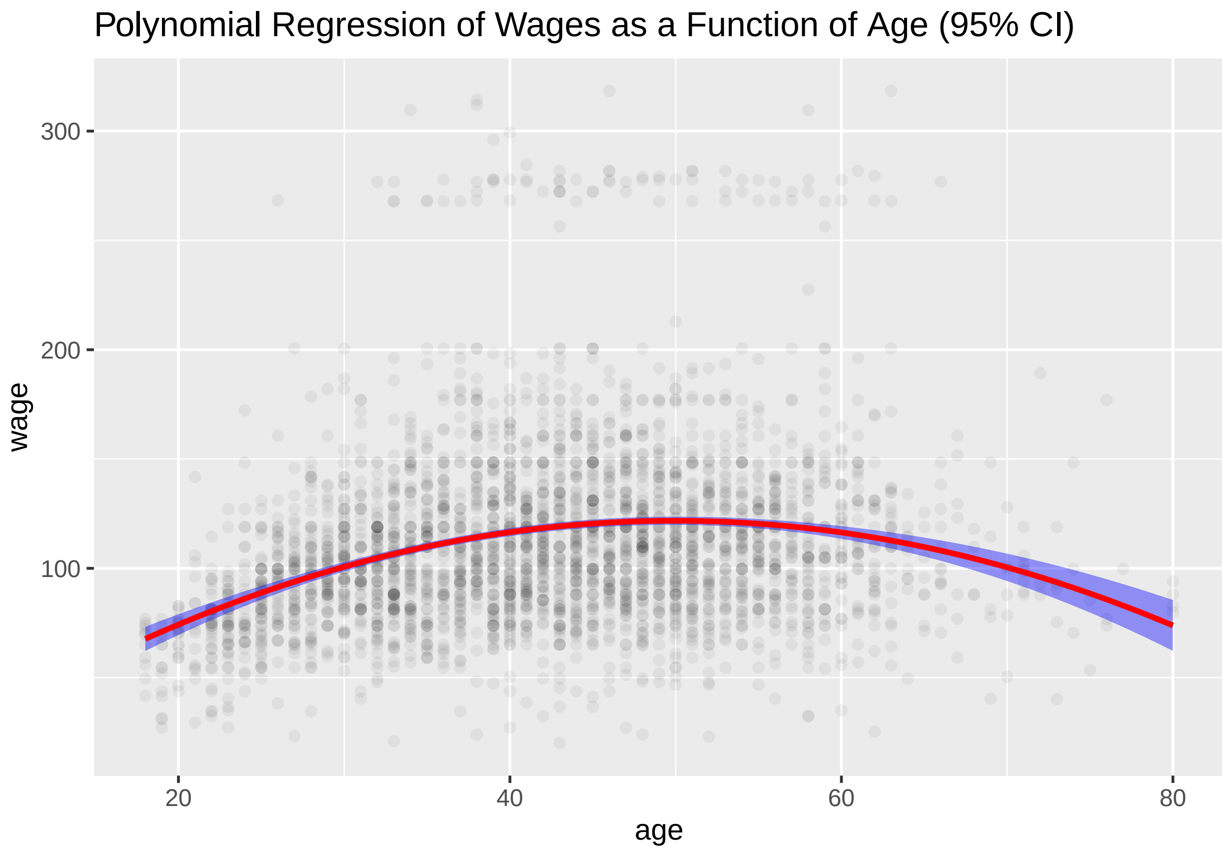
We get the following regression model (where X_1 represents age, and X_2 age squared).

$$\hat{Y} = -10.43 + 5.29X_1 - 0.05X_2$$

This model yields an adjusted r-squared of 0.08, suggesting that it is not a great fit. While the intercept term is not interpretable, we see in that age is positively correlated with wage. Specifically, for every year older one individual is than another, we expect an increase in wages by 5.3 (thousand) dollars.

(b)

```r
ggplot(wages,aes(y=wage,x=age),col="black") +
    geom_point(alpha=0.05)+stat_smooth(method="lm",formula = y~poly(x,2),fill="blue",col="red") +
    ggtitle("Polynomial Regression of Wages as a Function of Age (95% CI)")
```



In the above plot, the blue area surrounding the red line represents the 95% confidence interval.

(c) As seen above, our model doesn't seem to be a great fit for the data. The data is somewhat stratified, with a large amount of ~$260k wages which our model doesn't account for. Further, the data is widely scattered around each age value, suggesting that perhaps we need additional features to more accurately predict wage. By fitting a polynomial regression, we are asserting that wage increases, then decreases (or vice versa), which is not an unreasonable initial guess. However, the data doesn't lend itself particularly well to a quadratic upon further inspection.

(d) Statistically, a linear model is of the form $Y = \beta_0 + \beta_1 X$ while a polynomial regression is of the form $Y = \beta_0 + \beta_1 X + \beta_2 X^2 + ... + \beta_n X^n$. The $X_1, X_2, ..., X_n$ can be either distinct features, or one feature that has been transformed (i.e. squared). Because of the higher order terms, a polynomial model can "bend", and is more flexible than a linear model, which resembles a straight line. However, increasing the order too much can lead to overfitting (an idea encapsulated by the "bias-variance tradeoff").