Code ▾

Machine Learning PSET 1

Krish Suchak

1. Statistical and Machine Learning

Diff between supervised and unsupervised learning in 500-800 words.

- What is the relationship between the X’s and Y?
- What is the target we are interested in?
- How do we think about data generating processes?
- What are our goals in approaching data?
- How is learning conceptualized? And so on. . .

Supervised Learning

In supervised learning, input / output paris (both X and Y) are provided. The constructed model infers a relationship between the two variables, independent and dependent. The existence of a given set of Y values guides the model’s learning, and we assume there is an explicit relationship between the provided X and Y.

We measure the fitness of our model by using a metrics of accuracy (bias and variance). Supervised learning problems fall into two main categories: regression and classification. Regression problems are ones where the output variable is numerical, a quantifiable value. Conversely, classification problems are ones where the output variable is a class / category, a qualitative label.

For regression, a commonly used metric for closeness of fit is R^2 or the coefficient of determination. The closer R^2 is to 1 in the interval $[0, 1]$, the greater capability of the model to explain variation in the y-data based on variation in the x-data.

For classification problems, we use an accuracy score and confusion matrix to assess the ratios of false positives, false negatives, true positives, and true negatives. Examples of popular supervised learning algorithms include linear regression for regression problems, random forest for both regression and classification problems, and support vector machines for classification problems.

Unsupervised Learning

In unsupervised learning, only no explicit $X \rightarrow Y$ mapping is provided. The constructed model infers relationships between memeber of the set of X values. The lack of an explicit correct answer (Y values in supervised learning) means we do not assume any explicit relationships in the data.

Unsupervised learning techniques fall into two main categories: clustering and dimensionality reduction. For cluster analysis, we wish to discover distinct, inherent groupings in the data (similar to classification except without the predefined labels). For dimensionality reduction, we wish to reduce the number of random variables under consideration by obtaining a set of principal variables (for example, ones responsible for the most variance in the data). One example of this is principal component analysis (PCA) - a dimensionality reduction technique that converts a set of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components such that the first component has the highest variance possible and each successive component has the variance possible under the condition that it is orthogonal to the previous components.

Since there are no output variables to compare with the results of these techniques, there is also no metric for accuracy. However, if one uses these techniques on data where Y values do exist, then one can find potentially useful features through PCA or potential groups for classification through clustering. K-means is an example of a clustering algorithm which works by initially selecting randomly placed cluster centers and assigning every point to the nearest center. In each iteration of the algorithm, the centers are moved to the average location of the point assigned to it. We repeat the last two steps (assigning data points to centers and moving cluster centers accordingly) until the cluster centers effectively stop moving.

Semi-Supervised Learning

In short, we generate a function that seeks to map some relationship $X \rightarrow Y$ in supervised learning and in unsupervised learning, we explore possible structure in the data without prior assumptions. Lastly, semi-supervised techniques (a mix of supervised and unsupervised techniques) prove especially useful for data that contains missing labels.

2. Linear Regression

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

Hide

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

```

Call:
lm(formula = mpg ~ cyl, data = mtcars)

Residuals:
    Min      1Q  Median      3Q 
-4.9814 -2.1185  0.2217  1.0717 
    Max 
 7.5186 

Coefficients:
            Estimate Std. Error
(Intercept)  37.8846     2.0738
cyl          -2.8758     0.3224
            t value Pr(>|t|)    
(Intercept)   18.27  < 2e-16 ***
cyl           -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
```

Hide

```
coefficients(simple_fit)
```

```
(Intercept)         cyl 
   37.88458    -2.87579 
```

- The output is a linear model fit to `cyl` as the independent variable and `mpg` as the dependent variable with an R^2 value of 0.72. This means that the model explains 72% of the variation in the y data as a result of variation in the x data. The parameter values are β_0 (y-intercept) and β_1 (independent variable slope). The negative slope implies a negative correlation between `cyl` and `mpg` .
    - $\beta_0 = 37.885$
    - $\beta_1 = -2.876$

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

- $Y = \beta_0 + \beta_1 X_1 + \epsilon \Rightarrow Y = 37.885 - 2.876 X_1,$
- where Y is dependent variable (`mpg`) and X_1 is independent variable (`cyl`)

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

Hide

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

```

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

Residuals:
    Min      1Q  Median      3Q 
-4.2893 -1.5512 -0.4684  1.5743 
    Max 
 6.1004 

Coefficients:
            Estimate Std. Error
(Intercept)  39.6863     1.7150
cyl          -1.5078     0.4147
wt           -3.1910     0.7569
            t value Pr(>|t|)    
(Intercept)  23.141  < 2e-16 ***
cyl          -3.636 0.001064 ** 
wt           -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
```

Hide

```
coefficients(multi_fit)
```

```
(Intercept)         cyl          wt 
  39.686261   -1.507795   -3.190972 
```

Hide

- $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \epsilon \Rightarrow Y = 39.686 - 1.508X_1 - 3.191X_2$,
  - where Y is dependent variable (`mpg`) and X_1 is independent variable (`cyl`) and X_2 is the independent variable (`wt`)
- Note that the greater absolute value for coefficient for `wt` (compared to that of `cyl`) means that a change (Δ) in `wt` (ΔX_2) corresponds to a greater part in predicting `mpg` than the same change in `cyl` (ΔX_1).
  - Also, note that $R^2 = 0.82$. This means that this multiple linear model fits `mpg` better (explains more of the variation in `mpg` data) than the simple model (based on only one independent variable) from parts a) and b).

d. 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?

Hide

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

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

Residuals:
    Min      1Q  Median      3Q 
-4.2288 -1.3495 -0.5042  1.4647 
    Max 
 5.2344 

Coefficients:
            Estimate Std. Error
(Intercept)  54.3068     6.1275
cyl          -3.8032     1.0050
wt           -8.6556     2.3201
cyl:wt        0.8084     0.3273
            t value Pr(>|t|)    
(Intercept)   8.863 1.29e-09 ***
cyl          -3.784 0.000747 ***
wt           -3.731 0.000861 ***
cyl:wt        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
```

Hide

```
coefficients(inter_fit)
```

```
(Intercept)         cyl          wt 
 54.3068062  -3.8032187  -8.6555590 
     cyl:wt 
  0.8083947 
```

Hide

- $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + \epsilon \Rightarrow Y = 54.307 - 3.803 X_1 - 8.656 X_2 + 0.808 X_1 X_2$,
  - where Y is dependent variable (`mpg`) and X_1 is independent variable (`cyl`) and X_2 is the independent variable (`wt`)
- Note that $R^2 = 0.86$ (marginally higher than the previous model), and that the y-intercept has increased drastically. `wt` is still the variable with the coefficient of the highest absolute value.
- By adding a multiplicative interaction, we are implicitly asserting that to some degree, there exists a dependent relationship between two variables (`cyl` and `wt`).

3. Non-Linear Regression

a. Fit a polynomial regression, predicting `wage` as a function of a second order polynomial for `age` . Report the results and discuss the output.

Hide

```
wage_data <- read.csv(file = 'wage_data.csv')
quad_fit <- lm(formula = wage ~ poly(age, 2, raw = TRUE), data = wage_data)
summary(quad_fit)
```

```


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
(Intercept)               -10.425224   8.189780
poly(age, 2, raw = TRUE)1   5.294030   0.388689
poly(age, 2, raw = TRUE)2  -0.053005   0.004432
                          t value Pr(>|t|)    
(Intercept)                -1.273    0.203    
poly(age, 2, raw = TRUE)1  13.620   <2e-16 ***
poly(age, 2, raw = TRUE)2 -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
```

Hide

```
coefficients(quad_fit)
```

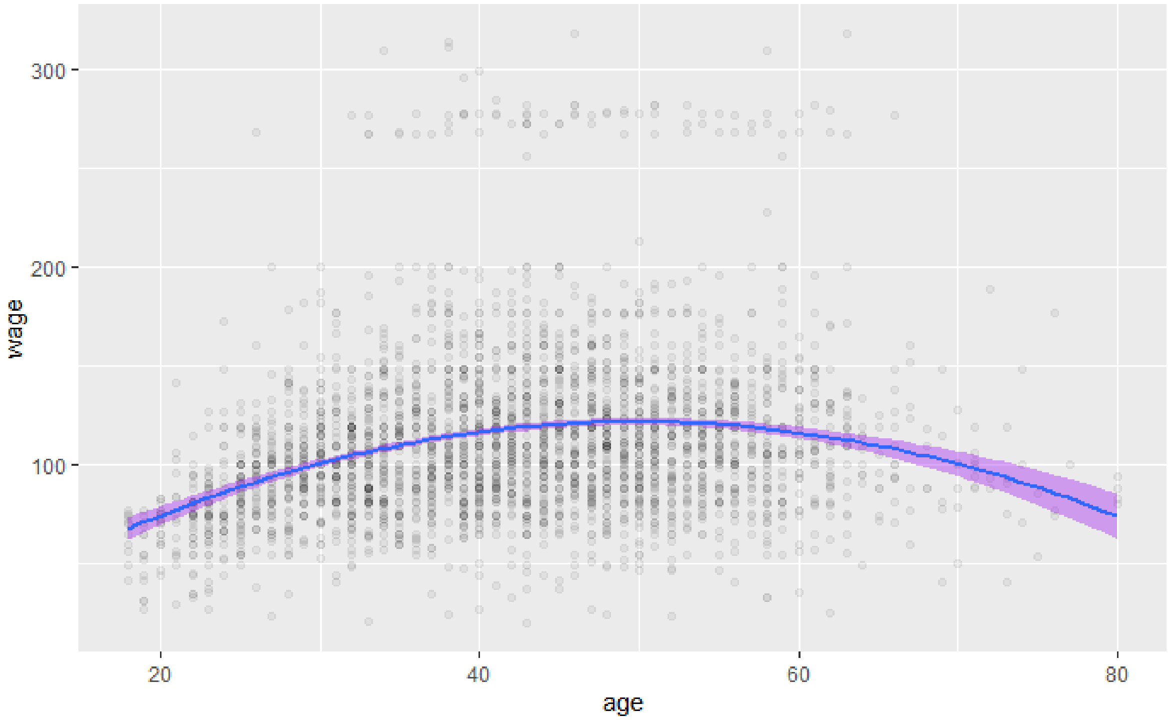
```
              (Intercept) 
             -10.42522426 
poly(age, 2, raw = TRUE)1 
               5.29403003 
poly(age, 2, raw = TRUE)2 
              -0.05300507 
```

- $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_1^2 + \epsilon \Rightarrow Y = -10.425 + 5.294 X_1 - 0.053 X_1^2$,
    - where Y is dependent variable (`wage`) and X_1 is independent variable (`age`)
- The fit has a low R^2 value (0.08). This means that not much of the variation in Y can be explained by variation in X.
- The quadratic fit has a concave shape since the quadratic coefficient is negative.

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

Hide

```
library(ggplot2)
ggplot(wage_data, aes(y = wage, x = age)) +
  geom_point(alpha = .05) +
  stat_smooth(method = "lm",fill = 'purple', formula = y ~ poly(x, 2))
```



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

- In the plot, we see a cluster of points between $X = 20$ and $X = 60$ and $Y = 50$ and $Y = 200$. The model clearly does not fit the data well, and the low R^2 value supports that view.
- By fitting a polynomial regression, we are asserting that Y grows or shrinks quickly for some values of X (non-constant slope). By specifically fitting a quadratic regression, we are asserting there is some global maximum or minimum where y values stop growing or stop diminishing toward.

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

- As mentioned in the last part, polynomial regressions substantively imply a non-constant relationship between X and Y with the existence of a curved function.
- Statistically, although polynomial regression fits a non-linear model to the data, it is still considered to be a linear problem for the sake of statistical estimation. Consequently, we treat polynomial regression as a special case of multiple linear regression.