IFT6758 Fall 2019 - Assignment 1

Part 2 - Function Fitting

Linear Regression

1. [ISLR 3.7.5] Consider the fitted values that result from performing linear regression without an intercept. In this setting, the i^{th} fitted value takes the form, $\hat{y}_i = x_i\hat{\beta}$ where
$$\hat{\beta} = \frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i^2}$$
Show that we can write
$$\hat{y}_i = \sum_{i'} a_{i'} y_{i'}.$$
What is $a_{i'}$? Note: We interpret this result by saying that the fitted values from linear regression are **linear combinations** of the response values.

Solution

We rearrange based on given information:
$$\hat{y}_i = x_i\hat{\beta} = x_i \frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i^2} = \frac{x_i}{\sum_{i=1}^{n} x_i^2} \sum_{i=1}^{n} x_i y_i$$
To simplify, we denote:
$$b_{i'} = \frac{x_i}{\sum_{i=1}^{n} x_i^2}.$$
Substitute back in:
$$\hat{y}_i = b_{i'} \sum_{i=1}^{n} x_i y_i = \sum_{i'} b_{i'} x_{i'} y_{i'}$$
Of which, we can further denote:
$$\hat{y}_i = \sum_{i'} a_{i'} y_{i'}$$
Whereby:
$$a_{i'} = b_{i'} x_{i'} = \frac{x_i}{\sum_{i=1}^{n} x_i^2} \sum_{i=1}^{n} x_i = \frac{\sum_{i=1}^{n} x_i}{\sum_{i=1}^{n} x_i^2} x_i$$

Extending Linear Regression

1. [ISLR 7.9.3] Suppose we fit a curve with basis functions $b_1\left(x\right) = x, b_2\left(x\right) = \left(x-1\right)^2 1\{x \geq 1\}$. (Note that $1\{x \geq 1\}$ equals 1 for $x \geq 1$ and 0 otherwise.) We fit the linear regression model,
$$y = \beta_0 + \beta_1 b_1\left(x\right) + \beta_2 b_2\left(x\right) + \epsilon$$
and obtain coefficient estimates
$$\hat{\beta}_0 = 1, \hat{\beta}_1 = 1 \hat{\beta}_2 = -2.$$
Sketch the estimated curve between $x = -2$ and $x = 2$. Note the intercepts, slopes, and other relevant information.

Solution

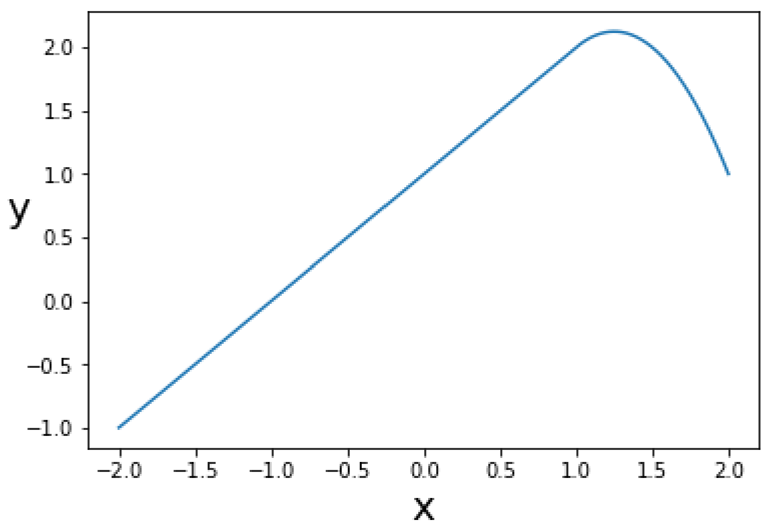
The plot is shown below. Some additional information follows suite.

In [0]:
```
import matplotlib.pyplot as plt
import numpy as np

# Initialize two numpy arrays, note y is only initiated with same len
gth and
# will be updated with correct values later to improve spatial comple
xity/cost
x = np.linspace(-2, 2, 1000)
y = np.zeros(1000)

# Update with proper values
for i in range(len(x)):
  # Implement the condition posed by the inary response function
  if x[i] < 1:
    y[i] = 1 + x[i] - 2 * (x[i] * x[i] - 2 * x[i] + 1) * 0
  else:
    y[i] = 1 + x[i] - 2 * (x[i] * x[i] - 2 * x[i] + 1) * 1


plt.plot(x,y)
plt.xlabel("x", fontsize = 20)
plt.ylabel("y", rotation = 0, fontsize = 20)
plt.show()
```



With the binary function, the model function is simplified to:

$$y = \begin{cases} 1+x, x \in [0,1) \\ -2x^2+5x-1, x \in [1,2] \end{cases}$$

From here, the ranges of the function for the given domain are:

$$y \in \begin{cases} [-1,2), \text{for } x \in [0,1) \\ [2,1], \text{for } x \in [1,2] \end{cases}$$

For the given range, the function's maximum value is ***2*** and the minimum is ***-1***

Intercepts

In the chosen range, the function crosses both axis exactly once. To solve for it, set x or y equal to zero, and for both parts, check if the corresponding y or x values fall within the range. For the given plot, x and y axis are crossed both exactly once:

$$\begin{cases} \text{x-intercept: } (-1,0) \\ \text{y-intercept: } (0,1) \end{cases}$$

Slopes

The slopes are just derivatives of the function at each range. The function is a straight line in the first part (between ***-2*** and ***1***), the slope is constant; the second part (between ***1*** and ***2***) is quadratic, the slope is a linear function:

$$y' = \begin{cases} 1, x \in [0,1) \\ -4x+5, x \in [1,2] \end{cases}$$

Other Information

> There is a knot at ***x = 1***

The function is increasing for the linear part (***-2*** and ***1***) and is decreasing for the second part (***-2*** and ***1***), by obvserving the plot.

Trees

1. [ISLR 8.4.4] Consider the figure below.

   a. Sketch the tree corresponding to the partition of the predictor space illustrated in the left-hand panel. The numbers inside the boxes indicate the mean of Y within each region.

   b. Create a diagram similar to the left-hand panel, using the tree illustrated in the right-hand panel. You should divide up the predictor space into the correct regions, and indicate the mean for each region.