Linear and Logistic Regression

David I. Inouye

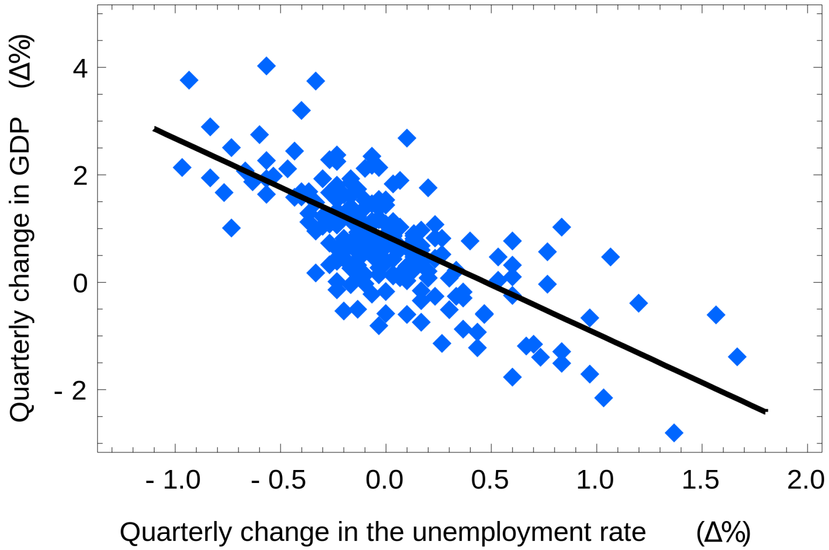
Tuesday, September 15, 2020

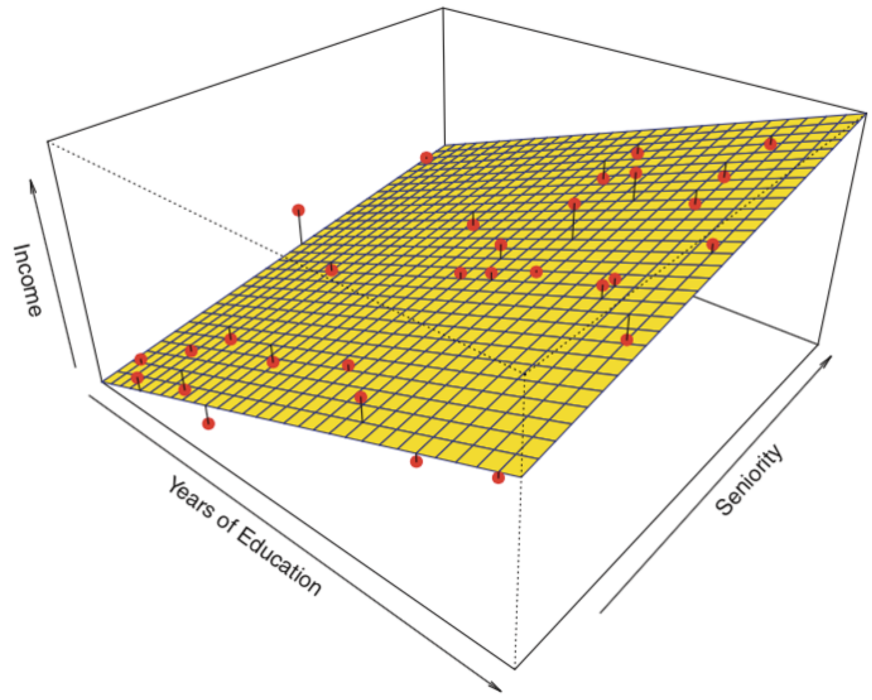
Outline

- Linear regression
  - Intuitions
  - Formalization
  - Solution in closed-form

- Logistic regression
  - Intuitions
  - Formalization
  - Solution requires numerical algorithms

Linear regression models the output as a line (1D) or hyperplane (>1D)





https://towardsdatascience.com/linear-regression-detailed-view-ea73175f6e86

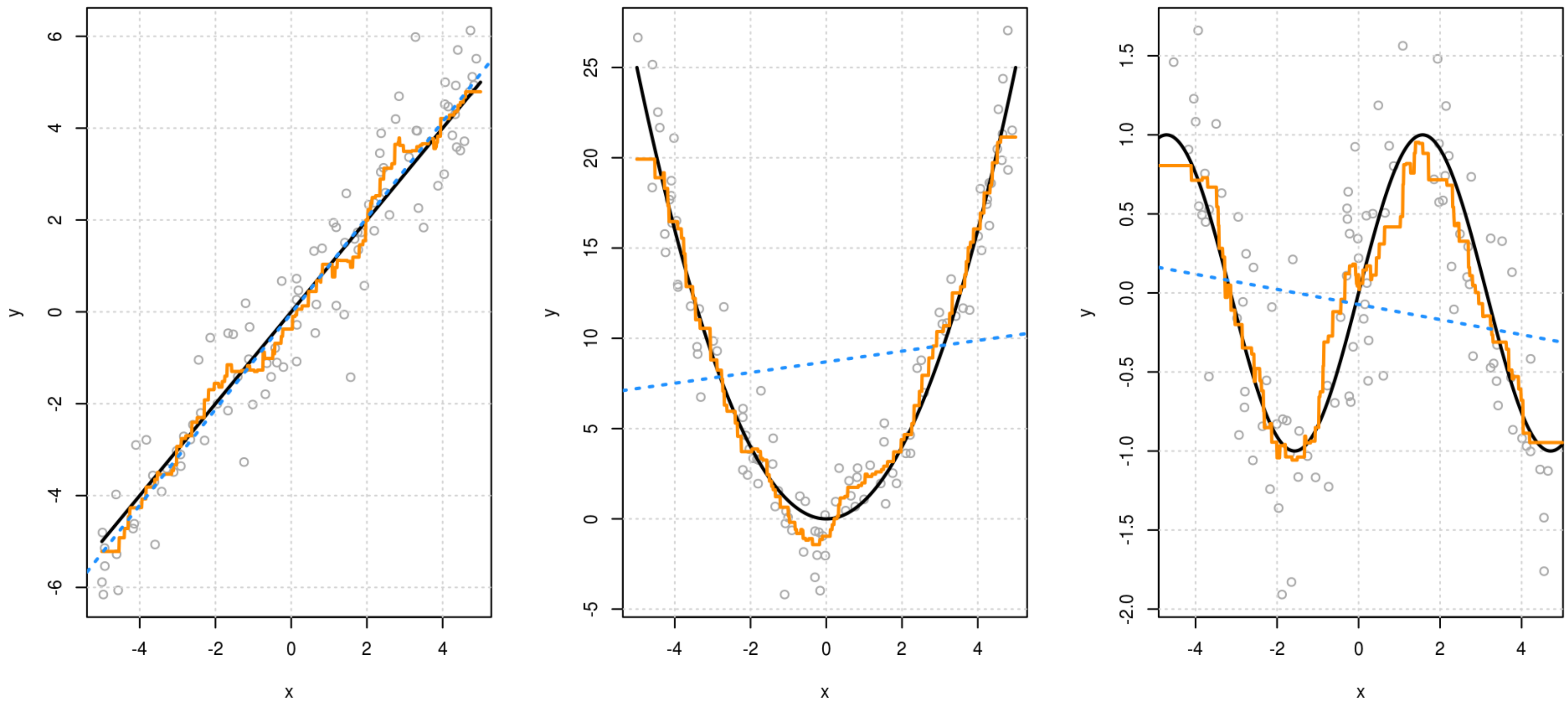
The linear regression model is defined by the coefficients (or **parameters**) for each feature

- A simple linear combination where θ are the parameters
$$f(x) = \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_d x_d + \theta_{d+1}$$

- Letting $\boldsymbol{x} = [x_1, x_2, \ldots, x_d, 1]$, we can write as
$$f(x) = \theta^T \boldsymbol{x}$$

- This is known as a **parametric model**

How does this compare to KNN regression? Linear regression is a much simpler function



https://daviddalpiaz.github.io/r4sl/knn-reg.html

The goal of linear regression is to find the parameters θ that minimize the prediction error

- Using mean squared error (MSE) this means:

$$\theta^* = \arg\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \big(y_i - f_\theta(\boldsymbol{x}_i)\big)^2$$

- Or equivalently

$$\theta^* = \arg\min_{\theta} \sum_{i=1}^{n} (y_i - \theta^T \boldsymbol{x}_i)^2$$

- Or in matrix form

$$\theta^* = \arg\min_{\theta} \|\boldsymbol{y} - X\theta\|_2^2$$

- Known as **Ordinary Least Squares (OLS)**

The solution for OLS can be computed in closed form

- How do you find maximum or minimum in calculus?
- Calculate gradient

$$\nabla_\theta \|\boldsymbol{y} - X\theta\|_2^2$$
$$= \left(2(\boldsymbol{y} - X\theta)^T(-X)\right)^T$$
$$= \left(2(-X^T)(\boldsymbol{y} - X\theta)\right)$$
$$= 2(-X^T\boldsymbol{y} + X^T X\theta)$$

Derivation hints: Use equivalence of $\|\boldsymbol{v}\|_2^2 = \boldsymbol{v}^T\boldsymbol{v}$. Then use [matrix calculus (wikipedia reference).]

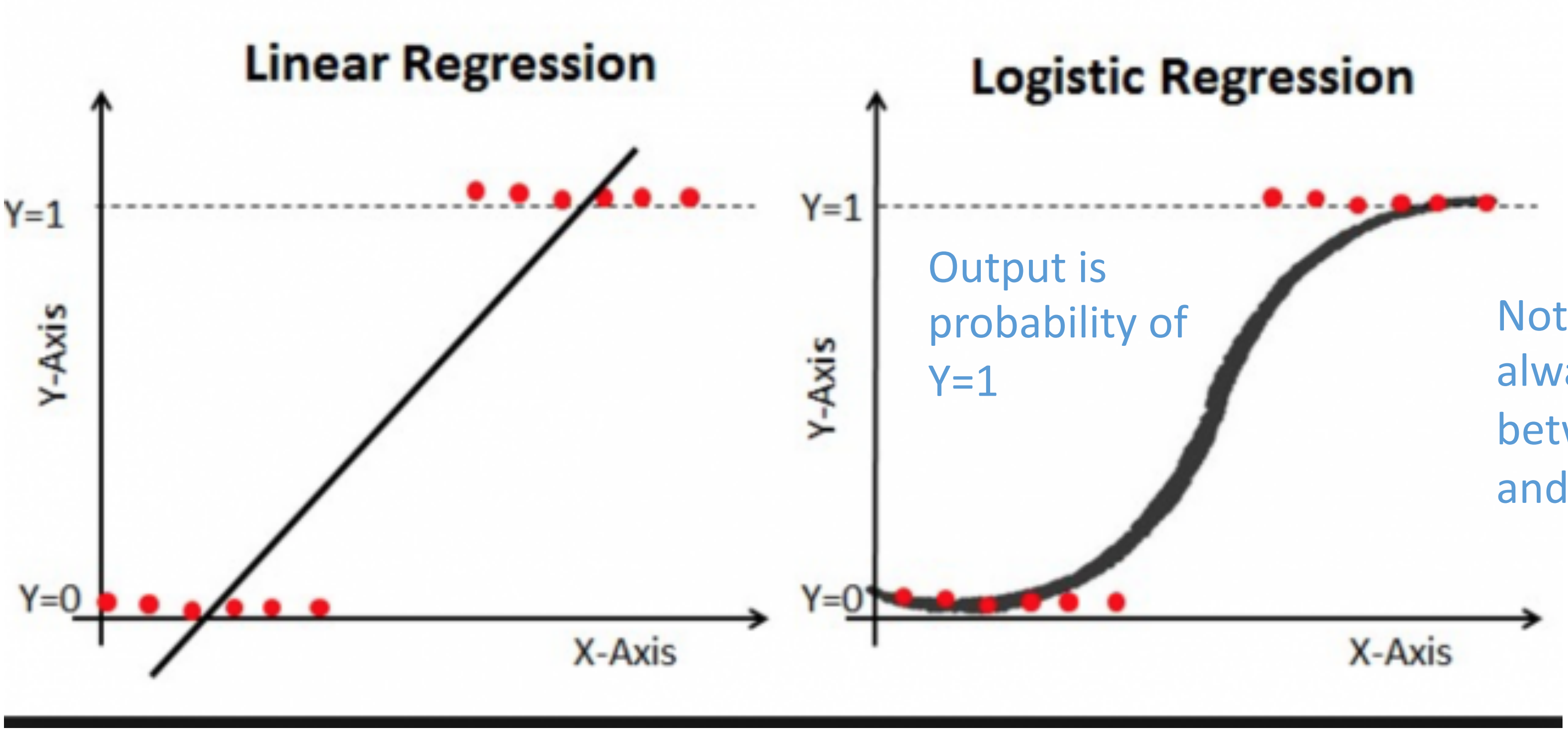
- Set equal to zero and solve

$$-2(X^T y + X^T X\theta) = 0$$
$$X^T X\theta = X^T\boldsymbol{y}$$
$$\theta^* = (X^T X)^{-1} X^T\boldsymbol{y}$$

Known as **normal equations**

https://en.wikipedia.org/wiki/Ordinary_least_squares

<u>Logistic regression</u> *generalizes* linear regression to the classification setting (*despite the name*)



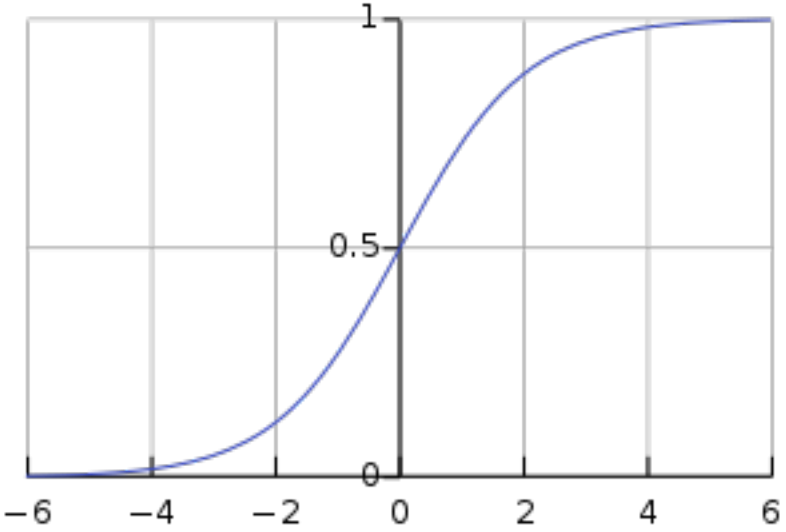
Output is probability of Y=1

Note that it is always between 0 and 1

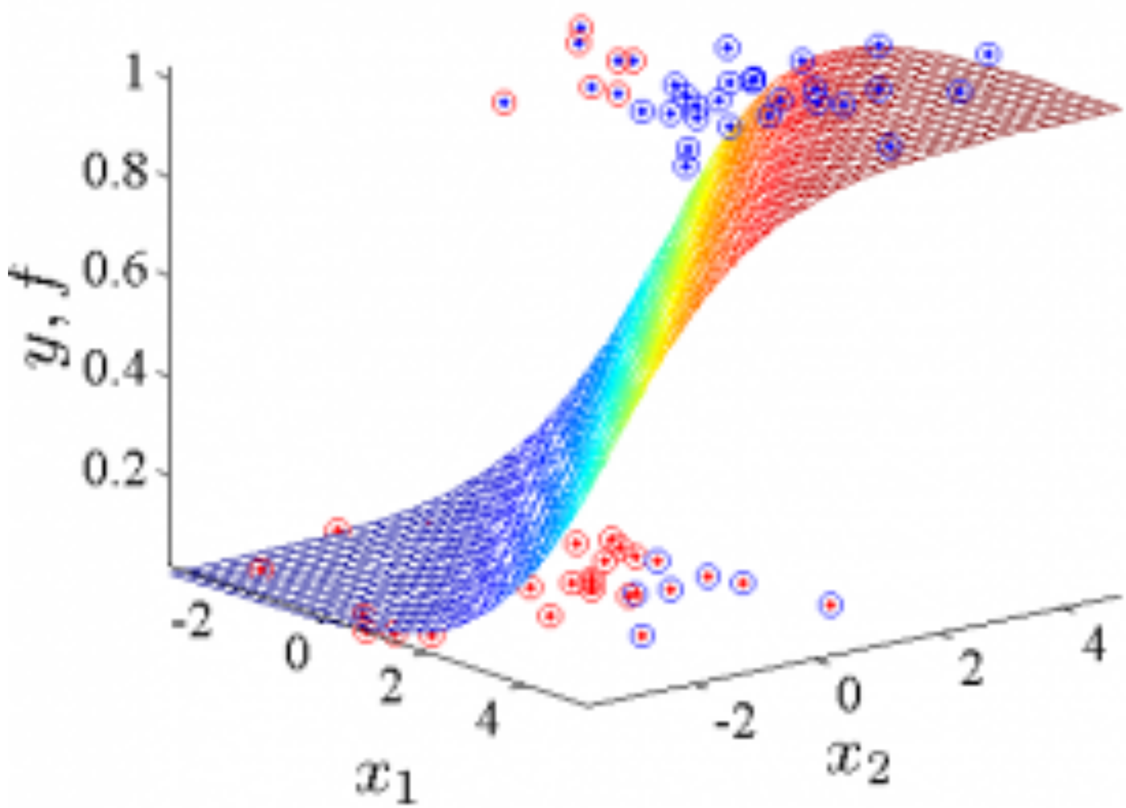
https://medium.com/@ODSC/logistic-regression-with-python-ede39f8573c7

The logistic function is a sigmoid curve with a simple form

- $\sigma(a) = \frac{1}{1+e^{-a}}$
- Equivalently
$$\sigma(a) = \frac{e^a}{e^a + 1}$$
- Bounds
  - $a \to \infty, \sigma(a) \to 1$
  - $a \to -\infty, \sigma(a) \to 0$
- 1D logistic model
$f(x) = \sigma(\theta_1 x + \theta_2)$



Logistic regression in higher dimensions is just the logistic curve **along a single direction**



Multivariate logistic regression merely applies a logistic function to the output of a linear function

- The multivariate logistic regression model is
$$f_\theta(\boldsymbol{x}) = \sigma(\theta^T \boldsymbol{x})$$
- Notice similarity to linear regression model
- However, we can *interpret* $f_\theta(x)$ as the **probability** of $y = 1$ instead of predicting y directly
- Thresholding this probability allows us to predict the class
$$\hat{y} = \begin{cases} 1, & \text{if } f_\theta(x) \geq 0.5 \\ 0, & \text{otherwise} \end{cases}$$

The logistic regression optimization minimizes the log likelihood of the training data

- In theory, we could use MSE:

$$\theta^* = \arg\min_\theta \|\boldsymbol{y} - \sigma(X\theta)\|_2^2$$

- However, the true output y is always 0 or 1
- Instead we maximize the **log likelihood** (which is equal to the log probability of the data)

$$\theta^* = \arg\max_\theta \sum_{i=1}^{n} y_i \log \Pr(y_i = 1|x_i) + (1 - y_i) \log \Pr(y_i = 0|x_i)$$

- Equivalently

$$\theta^* = \arg\max_\theta \sum_{i=1}^{n} y_i \log \sigma(\theta^T \boldsymbol{x}_i) + (1 - y_i) \log\big(1 - \sigma(\theta^T \boldsymbol{x}_i)\big)$$

Logistic regression does **not** have a closed-form solution!

- Must resort to numerical optimization
- Examples: Gradient descent, Newton's method