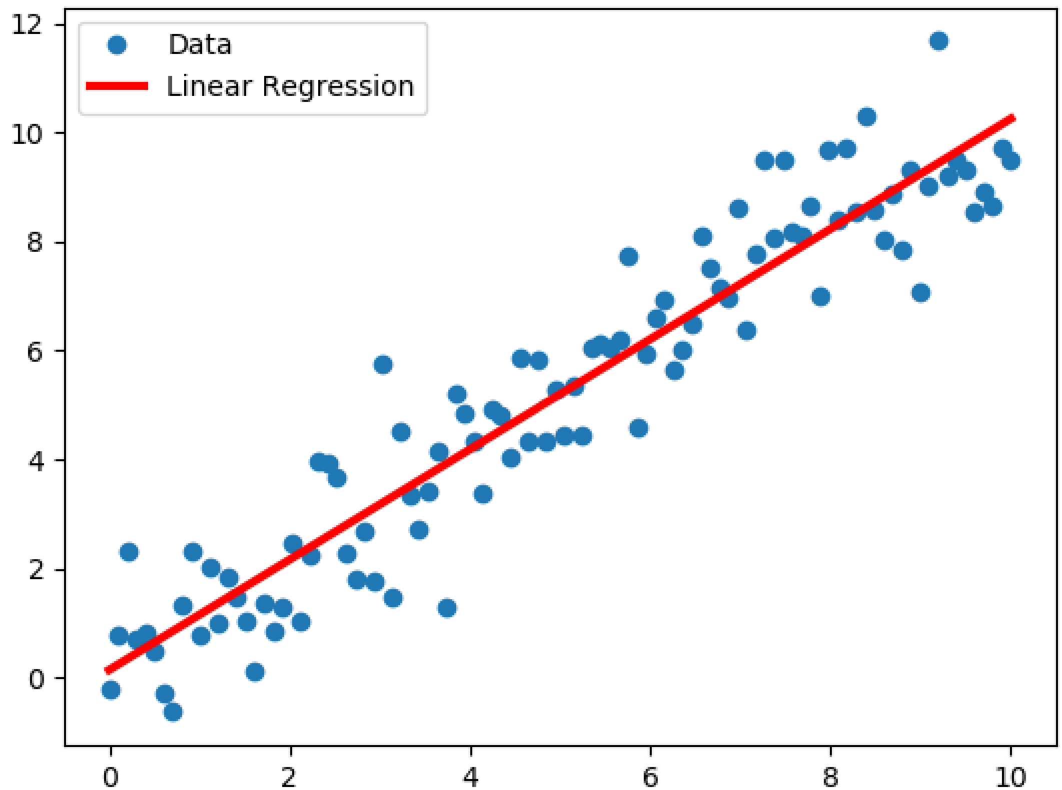
Optimisation

Motivation



We want to find a and b which minimize the sum square errors J which is given by

$$J(a, b) = \sum_{i=1}^{n} (y_i - (ax_i + b))^2$$

Next

We are going to find a and b which minimize $J(a, b)$ in three different ways:

- Numerically with Excel + Solver
- Analytically with Python Numpy
- Gradient Descent (will be calculated by you)

Numerically with Excel

- One variable linear regression

Matrix Representation

Recall that we would like to minimize

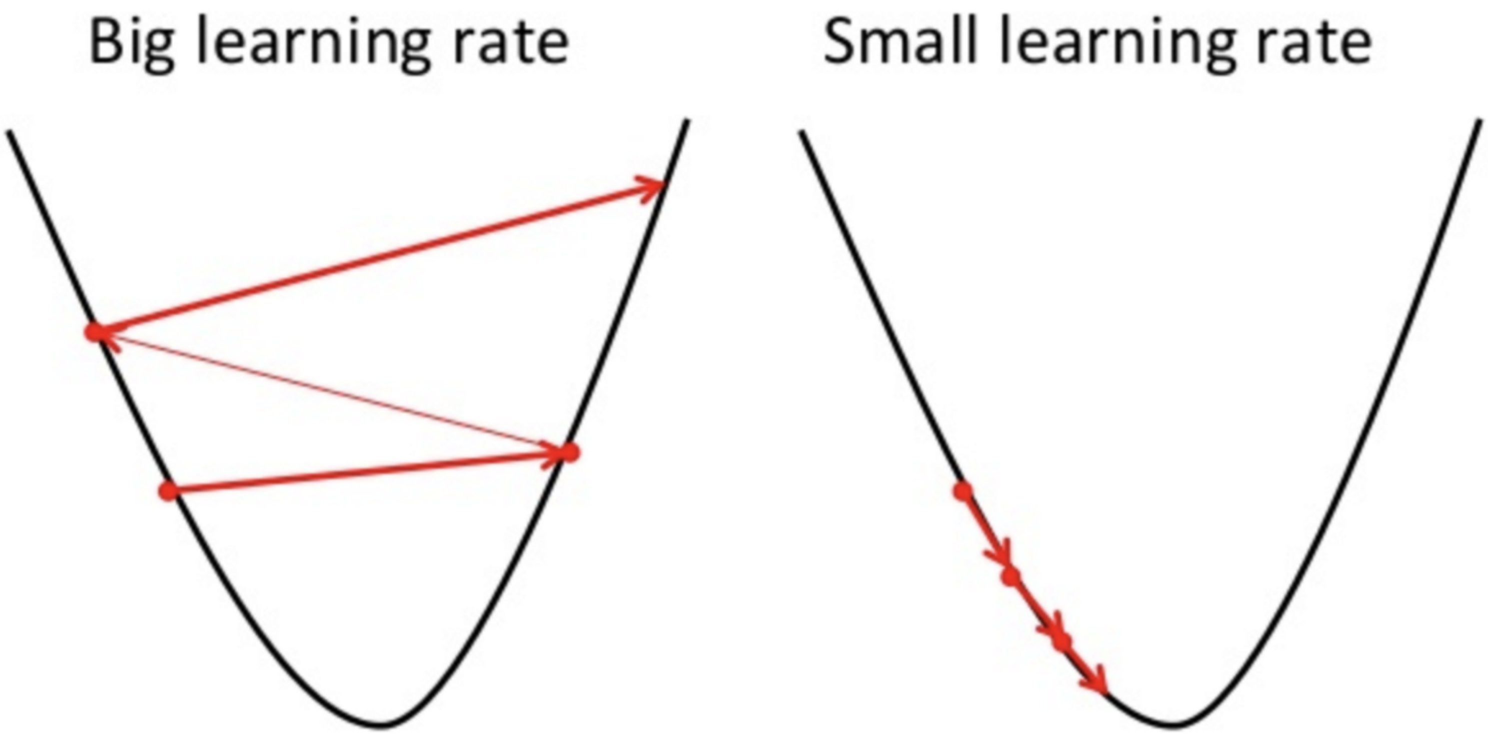
$$J(a, b) = \sum_{i=1}^{n} (y_i - (ax_i + b))^2$$

Suppose that we have n observations and m features. We can stack these observations in a matrix X with size $n \times m$.

If $J(\beta) = (y - X\beta)^T(y - X\beta)$, then $\hat{\beta}$ which minimize J is given by

$$\beta = (X^T X)^{-1} X^T y$$

Gradient Descent - Intuition



Gradient Descent - algorithm

For one variable, the iteration logic is given by:

$$x_{n+1} = x_n - \eta Df(x_n)$$

where $Df(x_0)$ means $\frac{df(x)}{dx}$ evaluated at $x = x_0$

If it is extended to multi-variable scheme, then the iteration logic becomes:

$$\beta_{n+1} = \beta_n - \eta \nabla J(\beta_n)$$

Remarks: η is called *learning rate*.

** Graph of J as function of a and b **

Differences between Convex vs Non-Convex

** Picture here **