Linear Regression with Single ~~Variable~~ feature

$$y_i = mx_i + c + \epsilon_i$$

MLE.

$$S(m, c) = \sum_{i=1}^{N} \epsilon_i^2$$

[LSE: Least Squares Error]

[Gives most optimal solution if ϵ_i is Gaussian.

If there are outliers, non-Gaussian dist. may be required.

$$= \sum_{i=1}^{N} (y_i - mx_i - c)^2$$

~~we~~ find m & c which minimise S.

Ex: House Price vs. Size.

$$\frac{\partial S}{\partial m} = 0 \Rightarrow -2\sum_{i=1}^{N} x_i(y_i - mx_i - c) = 0$$

$$\Rightarrow m\sum_{i=1}^{N} x_i^2 + c\sum_{i=1}^{N} x_i = \sum_{i=1}^{N} x_i y_i$$

$$\frac{\partial S}{\partial c} = 0 \Rightarrow -2\sum_{i=1}^{N} (y_i - mx_i - c) = 0$$

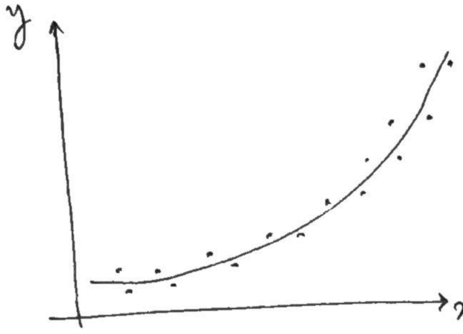
$$\Rightarrow m\sum_{i=1}^{N} x_i + cN = \sum_{i=1}^{N} y_i$$

Solving the simultaneous equation, we get

$$m = \frac{N\sum_i x_i y_i - \sum_i x_i \sum_i y_i}{N\sum_i x_i^2 - \left(\sum_i x_i\right)^2}$$

Linear Reg is one of the most popular & effective methods used in science & engineering. "Work Horse"

$$c = \frac{\sum_i y_i - m\sum_i x_i}{N}$$



$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2$

Can still be solved using Linear Regression methods.

$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2}$

where $x_{i1} = x_i$

$x_{i2} = x_i^2$

In general,

$$f(x) = \beta_0 + \sum_{j=1}^{m} x_j \beta_j \quad ; \quad X = [x_1, x_2, \ldots, x_m]$$

$$= \sum_{j=0}^{m} x_j \beta_j \quad \text{where } x_0 = 1.$$

Multiple Linear Regression

~~or Multivariate~~

Multivariate (General)

↳ multiple output.

$y = w_0 + w_1 x_1 + w_2 x_2 + \epsilon$

Five ~~two~~ data points : $y_i = w_0 + w_1 x_{1i} + w_2 x_{2i} + \epsilon_i \; ; \; 1 \le i \le 5$

$$X = \begin{bmatrix} 1 & x_{11} & x_{21} \\ 1 & x_{12} & x_{22} \\ 1 & x_{13} & x_{23} \\ 1 & x_{14} & x_{24} \\ 1 & x_{15} & x_{25} \end{bmatrix} \qquad w = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix}$$

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_5 \end{bmatrix}$$

MULTIPLE ~~MULTIVARIATE~~ LINEAR REGRESSION

Linear Regression with single feature: MLE with Gaussian error.

$$y_i = mx_i + c + \epsilon_i$$

m, c obtained by minimizing $S = \sum_i (y_i - mx_i - c)^2$

LSE.

Multiple LinReg:

$$y_i = w_0 + w_1 x_{1i} + x_{2i} + x_{3i} + \dots + w_m x_{mi} + \epsilon_i$$

$$= w^T x_i + \epsilon_i$$

$\rightarrow$ w & x_i are vectors instead of scalars.

but y is scalar. ~~y can also be a vector. (in which case w is a tensor)~~

$$S = \sum_i (y_i - w^T x_i)^2$$

~~or the generalised to the case of y being vector~~

$$= (y - Xw)^T (y - Xw)$$

vector of all data; vector; Matrix

Each x_i can be a non-linear function of other variables. So, this linear reg. can actually model polynomial & other functions as well.

$$= (y^T - w^T X^T)(y - Xw)$$

$$= y^T y - 2y^T Xw + w^T X^T X w$$

$$\nabla_w S = 2X^T Xw - 2X^T y = 0$$

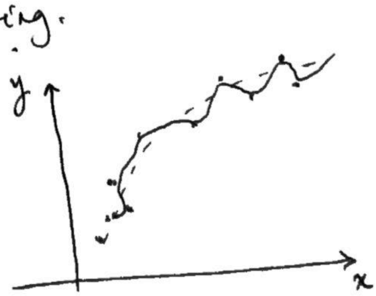
$$\Rightarrow \boxed{X^T X w = X^T y} \equiv \text{NORMAL EQUATION}$$

ORDINARY LEAST SQUARES

$$\Rightarrow w_{OLS} = (X^T X)^{-1} X^T y$$

$\rightarrow$ very hard to compute

If model complexity is large, parameter values can be large leading to unstable solution, i.e. w_{OLS} can change a lot for small changes in data points. Also leads to over-fitting.



USE REGULARISATION.

RIDGE REGRESSION

Encourage parameter values to be small [MAP]

use zero-mean Gaussian PRIOR

$$S = \frac{1}{N}\sum_{i=1}^{N}(y_i - w^T x_i)^2 + \lambda \underbrace{\|w\|_2^2}_{\rightarrow\, w^T w} \quad L_2 \text{ Regularisation term or weight decay.}$$

$$w_{RIDGE} = (\lambda I_D + X^T X)^{-1} X^T y$$

$\lambda = 0 \Rightarrow$ over-fitting

$\lambda \approx 1 \Rightarrow$ under-fitting.

$\|w\|_2 = \sqrt{w_0^2 + w_1^2 + w_2^2 + \cdots}$ — w_0 usually omitted from regularisation

LASSO

Use L_1 regularisation

$$S = \frac{1}{N}\sum_{i=1}^{N}(y_i - w^T x_i)^2 + \lambda \underbrace{\|w\|_1}_{L_1 \text{ norm}}$$

useful when we have outliers (non-Gaussian errors)

$\|w\|_1 = |w_0| + |w_1| + |w_2| + \cdots$

SUBSET SELECTION

Pick K best features using some thresholds.

BAYESIAN Linear Regression

Estimate errors in estimation of weights.

MULTIVARIATE (GENERAL) LINEAR REGRESSION

~~Multiple~~ Vector output

Same expression for Normal Eqn. works but y is now a matrix instead of a vector.