Supervised $\quad \mathcal{D} = \{x_i, y_i\}_{i=1}^{N}$

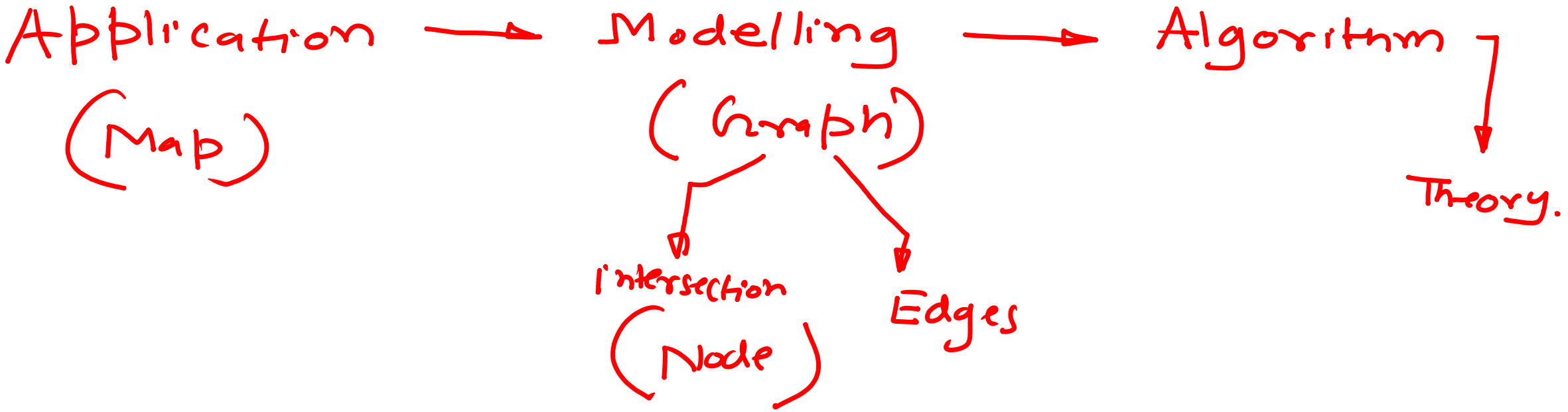
- Regression → $Y \in \mathbb{R}$
- Classification →
  - Fixed number of classes
  - obj :- to predict $\tilde{x}$ belongs to which class.

Obj :- We wants to learn the relation between x and y.

$$y = f(x)$$

↳ we want to find f.

* 4 steps for applying AI for any real-life problem.



* Linear Regression :-

Suppose $\mathcal{D} = \{x_i, Y_i\}_{i=1}^{N}$

X_1	X_2	X_3	Y
$x_1^{(1)}$	$x_2^{(1)}$	$x_3^{(1)}$	$y^{(1)}$
$x_1^{(2)}$	$x_2^{(2)}$	$x_3^{(2)}$	$y^{(2)}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$x_1^{(N)}$	$x_2^{(N)}$	$x_3^{(N)}$	$y^{(N)}$

We want to learn

$$Y = f(X_1, X_2, X_3)$$

* In linear regression,

$$Y = a_0 + a_1 X_1 + a_2 X_2 + a_3 X_3 + a_4 X_1 X_2 - (1)$$
$$+ a_5 X_1 X_3 + a_6 X_2 X_3 + a_7 X_1^2 + a_8 X_2^2 + a_9 X_3^2$$

In a generic form,

$$Y = a_0 \phi_0(X) + a_1 \phi_1(X) + \cdots + a_K \phi_K(X) \quad -(2)$$

Comparing (1) and (2)

$$\phi_0(X) = 1 \qquad \phi_2(X) = X_2$$

$$\phi_1(X) = X_1$$

* Eq. (2) is the generic expression of Linear regression

$$Y = \sum_{i=0}^{K} a_i \phi_i(X) \quad \text{---} \quad (2)$$

K ← number of basis func.

a_i ↑ unknown coefficients.

$\phi_i(X)$ ↑ basis function

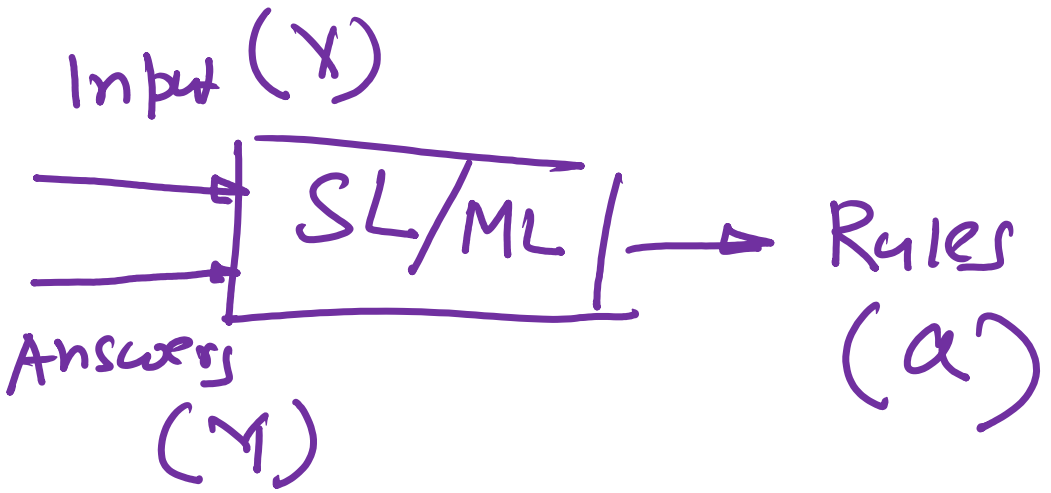
⚠ For using Linear regression in practice :-

Hyperparameter

(1) we need to fix K

(2) we need to select ϕ_i $\forall i$

* Learning in linear regression means computing the unknown parameters from data.



* Training :- (1) $\mathcal{D} = \{x_i, y_i\}_{i=1}^{N}$

(2) Hyper parameters.

* For a "good" value of a

$\tilde{y} = \sum_i a_i \phi_i(x)$ and the measured

y should ~~not~~ be close.

↳ we need to define a

error metric that measures

the distance between y, $\tilde{y}$

$E(a)$

* We find the best a by minimizing the error

$$a^* = \arg\min_a E(a)$$

* Some examples of error metric are :-

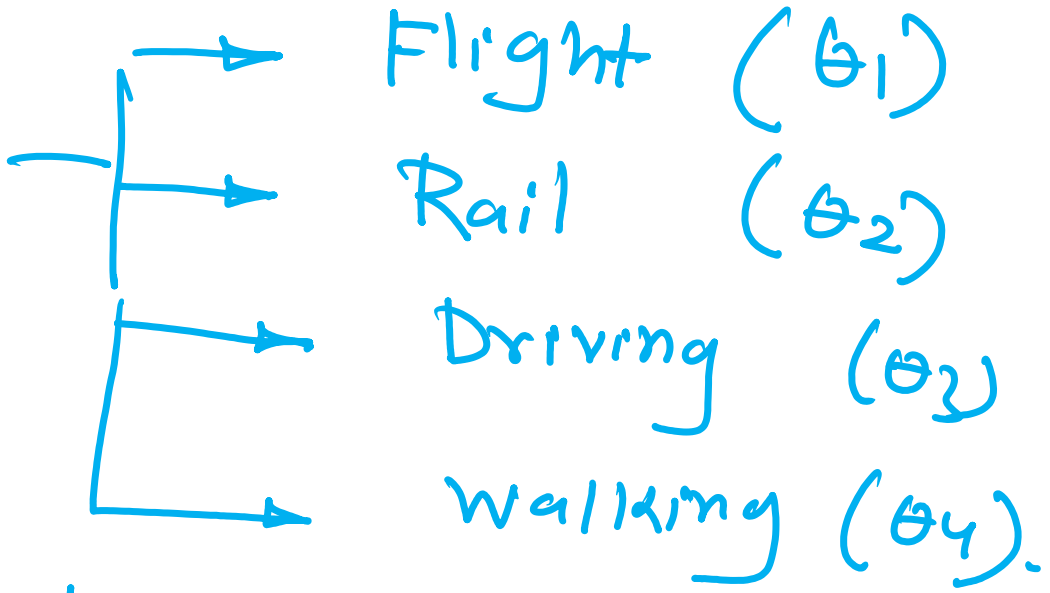
(1) Mean - squared error (MSE)

(2) Mean- absolute error (MAE)

* Traveling from Delhi to Kolkata.

* It took 24 hours. (x)

* Modes of travel



$P(x|\theta_1) = 0.1$

$P(x|\theta_2) = 0.9$

$P(x|\theta_3) = 0$

$P(x|\theta_4) = 0$

By generalizing,

$$\theta^* = \arg\max P(\mathcal{D}|\theta)$$

Maximum Likelihood Estimation.

* For linear regression, we have closed form solution.

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \; E(a)$$

$$\frac{dE(a)}{da} = 0 \Bigg\} \longrightarrow \text{System of equation}$$

* But often closed form solutions are not available.

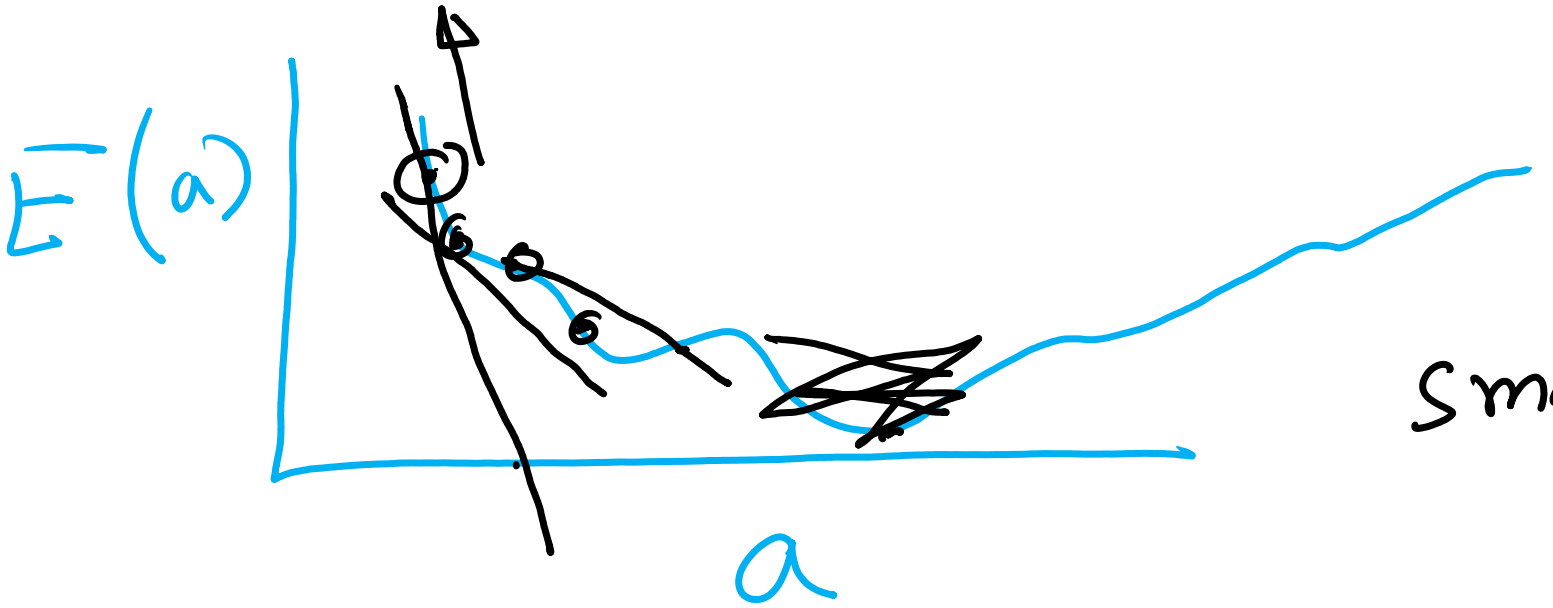
↳ we use numerical optimization

* One such numerical optimization algorithm is Gradient descent.

$$a^{(t+1)} = a^{(t)} - \eta \nabla_a E(a)$$

$\nabla_a E(a)$ ← Gradient

η ← learning rate



Small η ← lots of iterations

* Gradient ← direction of movement

large η ← oscillation

(*') η ← how much move

* Special case :- Linear regression with least squared loss function has a closed form solution.

$$\tilde{y} = \sum_{i=0}^{K} a_i \phi_i(x)$$

$$[y]_{N\times 1} = \phi_{N\times(K+1)}\, a_{(K+1)\times 1}$$

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} \phi_0(x^{(1)}) \end{bmatrix}$$

$$\begin{bmatrix} \phi_0(x^{(1)}) & \phi_1(x^{(1)}) & \phi_2(x^{(1)}) \cdots & \phi_K(x^{(1)}) \\ \phi_0(x^{(2)}) & \phi_1(x^{(2)}) & \phi_2(x^{(2)}) \cdots & \phi_K(x^{(2)}) \\ \vdots & & & \\ \phi_0(x^{(N)}) & \phi_1(x^{(N)}) & \phi_2(x^{(N)}) \cdots & \phi_K(x^{(N)}) \end{bmatrix}$$

$N \times (K+1)$

$$Y = \underline{\Phi}\, a$$

$$a = \left(\Phi^T \Phi\right)^{-1} \Phi^T Y$$

Algorithm

Input :- $\mathcal{D} = \{x_i, y_i\}_{i=1}^{N}$, K, Φ ∀ 1

(1) Formulate matrix Φ

(2) Form Y

(3) $a^* = (\phi^T \phi)^{-1} \phi^T y$

Return a^*

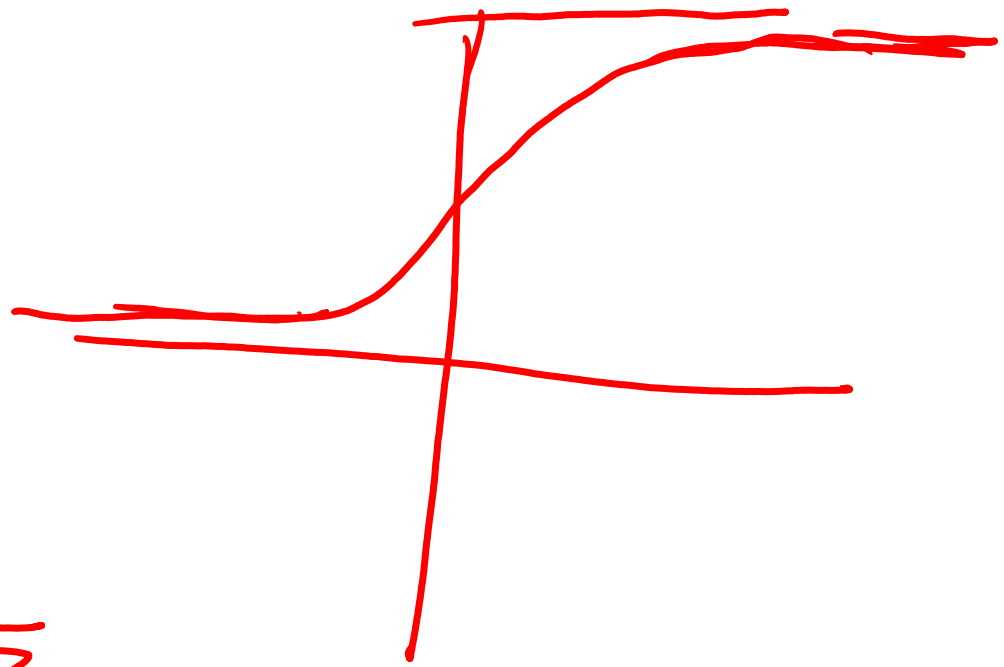
* Logistic Regression

Logistic regression is an algorithm for solving classification problems. (binary)

In logistic regression, the hypothesis is

$$f_\theta(x) = \sigma(ax)$$

"Sigmoid"



$$\sigma(z) = \frac{1}{1+e^{-z}}$$

Since we are dealing with binary classification $y \in \{0, 1\}$

$$P(y=1|x) = f_\theta(x) = \sigma(\underline{a}x) \rightarrow \sigma\left(\sum_i a_i x_i\right)$$

$$P(y=0|x) = 1 - f_\theta(x)$$

$$P(y_i|x_i) = f_\theta(x)^{y_i}\left(1 - f_\theta(x)\right)^{1-y_i}$$

$$P(\mathcal{Y}|\theta) = P(Y|X;\theta) = \prod_{i=1}^{N} P(y_i|x_i;\theta)$$

Assumption :- IID assumption

I → Independent

I → Identical

D → Dist

* In logistic regression, gradient descent has to be used as there exist no closed form solution.

Algorithm :- Logistic Regression

Input = $\mathcal{D} = \{x_i, y_i\}_{i=1}^{N}$

Initialize $a^{(1)} = a_0$.

for t = 1 : epoch

$$a^{(t+1)} = a^{(t)} + \eta \nabla_a P(y/a)$$

end

Return $a^{(t+1)}$

* In practice, we work with the log likelihood $\log P(x|a)$.

$$a^{(t+1)} = a^{(t)} + \eta \nabla_a \log P(x|a)$$