"Can machines think?"

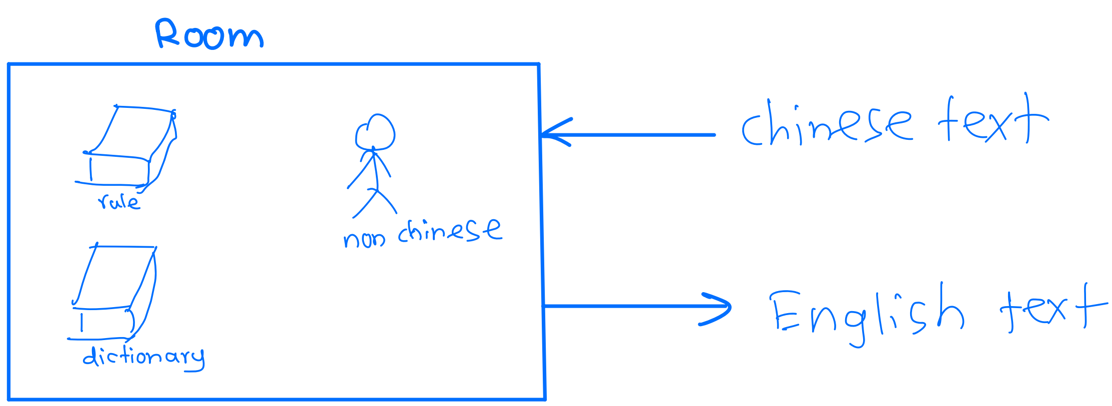
Alan turing assumed we can build intelligent machines.

Measuring intelligence is what he is proposing.

Turing anticipated all AI fields.

- Searching
- Reasoning (Reason based on knowledge that we have aquired)
- Knowledge representation
- NLP
- Computer vision
- Learning

The Chinese Room (J.R Searle)



Can this person understand chinese?
Can the room understand chinese?

(stupid)
Static/specific/explicit (directly stated) Tic-Tac-Toe

Version : Lookup table
↓
Answer for every situation is hard coded.

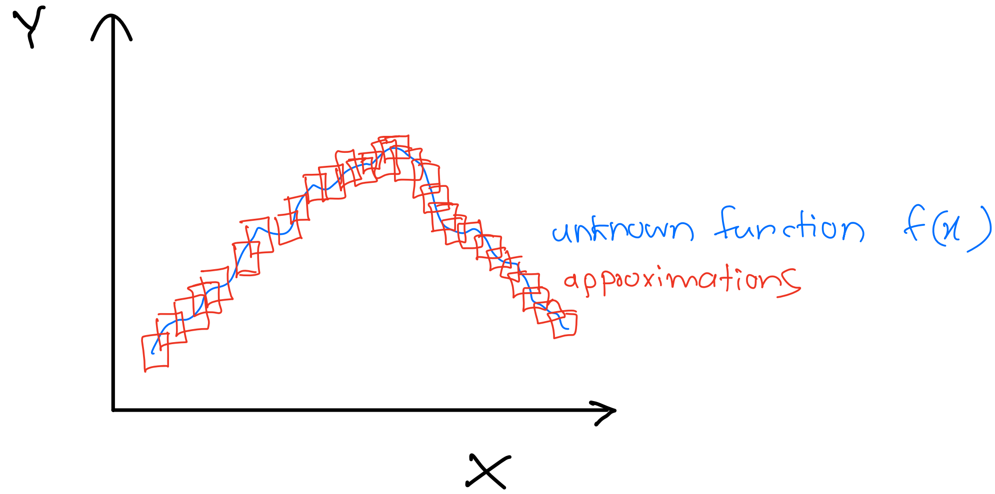
Version 2: A - attempt to place two marks in a row
B - if a opponent has two marks in a row, then mark the 3 spot.

Version 3: Represent the state of the game
- Current board position
- Next legal moves

Use an evaluation function
- Rate the next move (How likely to win)
- Look ahead if possible to see estimate opponent's moves

dynamic/generic/implicit
(intelligent)

AI → function approximation



We need data for any approximation

example : Linear Regression

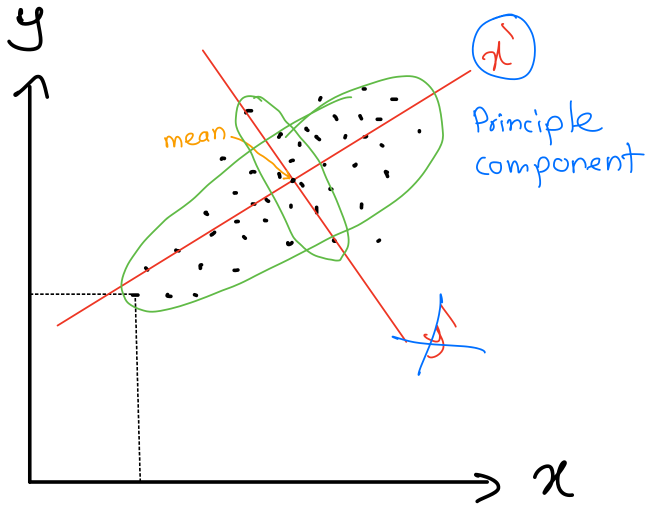
AI = we operate intelligently on data to extract the relationship between in- and outputs

→ Need training data for learning

→ Testing data for testing

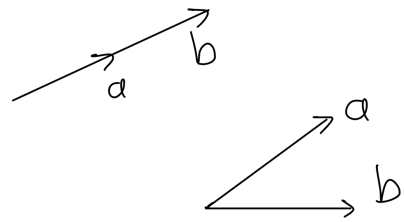
Problems ① Not enough data
→ augmentation

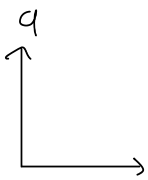
② Too much data
→ dimentionality reduction
↓
sometimes not every



data point is not valueble

correlated vectors



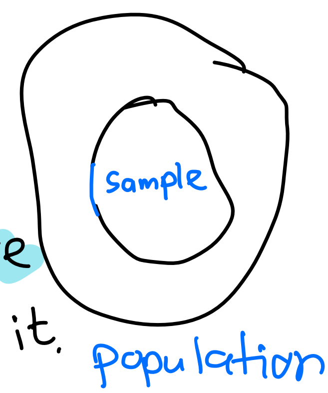


not correlated (orthogonal)

Desired: No correlation
High variance

x' gives more correlation

x' passes through the mean and diliver maximum variance when samples are projected into it.



Process is repeated until we find n such axes $\langle x_1, x_2, x_3 \dots x_n \rangle$

$x \in \mathbb{R}^d \longrightarrow n \leq d$

Principal Component Analysis (PCA) is algorithm to find orthogonal axes that diagonalize the covariance matrix.

x scalor

$\mathbf{x}$ vector

X set

$\mathbf{X}$ matrix

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$ n-dimentional input

$$\mathbf{x} = [x_1 \; x_2 \; \dots \; x_n]^T$$

Covariance Matrix $\Sigma = \text{cov}(x_i, x_j)$

ith input

j th input

$$= E[(x_i - \mu_i)(x_j - \mu_j)]$$

where $\mu_i = E[x_i]$

$\mu_j = E[x_j]$

expected value ~~average~~

Are x_i, x_j changing together?

(correlated?)

Generalization of covariance

$$\Sigma = E[(\mathbf{x} - E(\mathbf{x}))(\mathbf{x} - E(\mathbf{x}))^T]$$