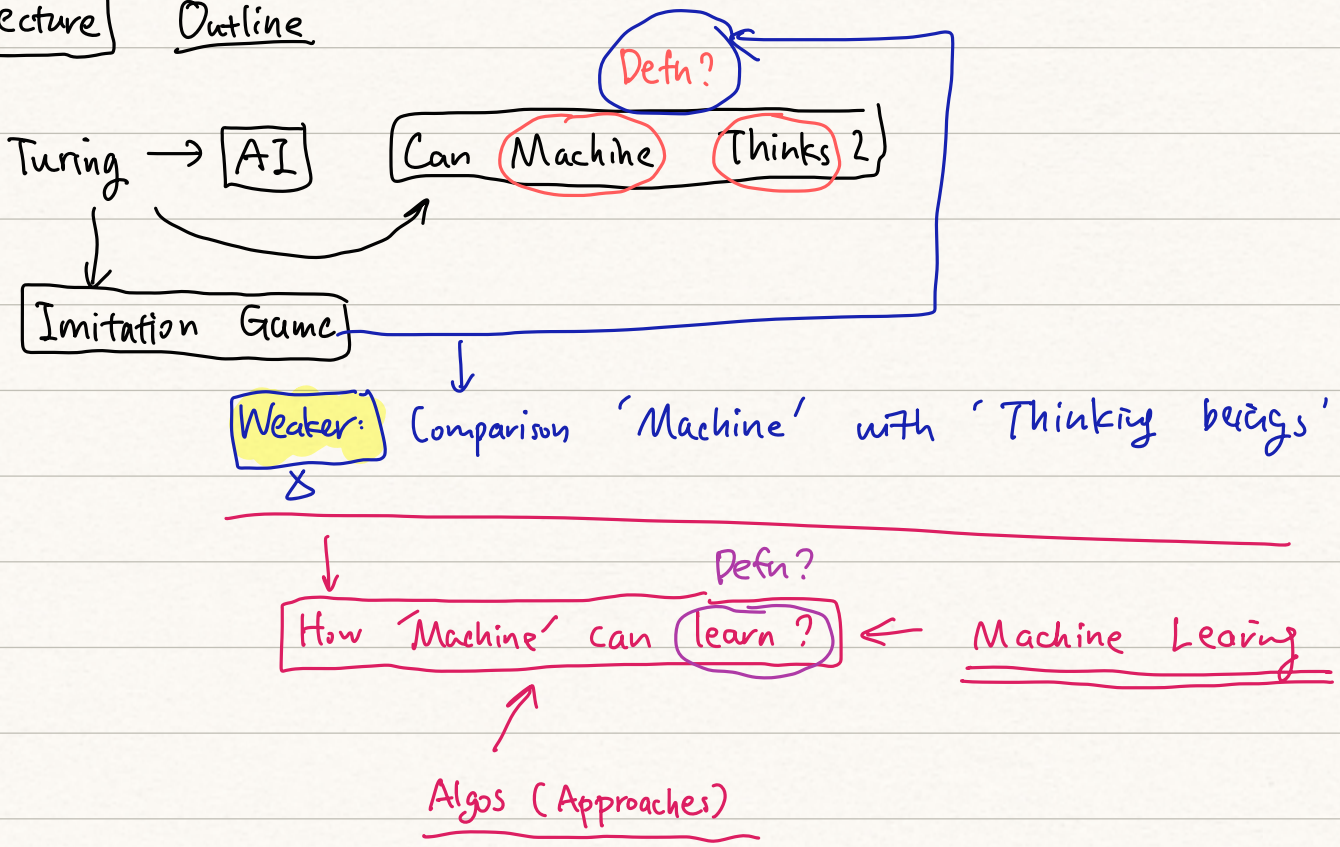
DSA5105 LEC1.

S17-05-08 ← office

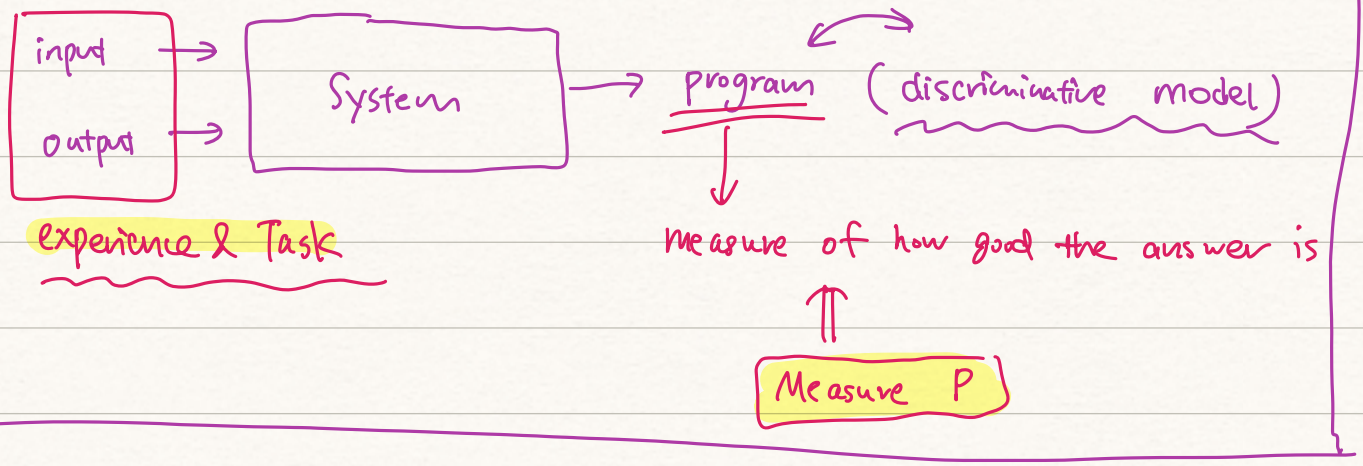
Lecture Outline



Learn Defn: { Experience E, Task T, Measure P

AI Learn Program from data

Example (Learning Process)



Outline

① Supervised ← { inputs, outputs } (labelled data)
- a) { classification, regression }
- b) Learning Theory
- c) Optimization algo ← SGD

② unsupervised ← only inputs (unlabelled data)
- a) dimension reduction (PCA)
- b) clustering
- c) generative model (GAN)

extract info from data

③ RL ← MDP

RL Algos

reward (state & action)

Numerical Data
- Grid-data
  - Images
  - Video
  - Time Series

Discrete Data ⇝ not $\mathbb{R}$, like $t \in \{1,2,3,4,5\}$

- ordinal data → have order. [e.g., level of proficiency] — preserve order
- Nominal Data (category data) → no order. [e.g., category like 'cat' & 'dog']
  - [e.g., true or false].
  - ↳ no natural data {pen, paper, bottle}
    - ↳ something symmetry within data structure while modelling
      - (one-hot encoding)
      - Should preserve with formulation

Different Formulation in application

inverse problem / design (?) ← Supervised learning

Notation: ① Dataset $\mathcal{D} = \mathcal{D}_{train} \cup \mathcal{D}_{test}$ → evaluate the performance of model

DATASET

② we can also split for one additional set $\mathcal{D}_{validation}$

$\mathcal{D} = \mathcal{D}_{train} \cup \mathcal{D}_{test} \cup \mathcal{D}_{validation}$

$\mathcal{D}_{validation}$: set for hyper-parameter tunning & Model Selection

- $\mathcal{D}_{train}$: determine the model for one specific hyperparameter
- $\mathcal{D}_{test}$: evaluate the performance
- $\mathcal{D}_{validation}$: Choose the optimal Model ← candidate of hyper-para

Framework:

parameters candidate → train — validation → Find the optimal one → evaluate on test set

When training set is BIG, validation set can be diminated. Do model selection on Traing set Directly

---

Supervised Learning Algo

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

Goal: learn relationship: $x_i \rightarrow y_i$ ↔ Oracle $f^*(\cdot)$

⇕

learn $\hat{f}^*(\cdot)$ s.t. $\hat{f}^*(x_i) = y_i$ $i = 1, 2, \ldots, N$

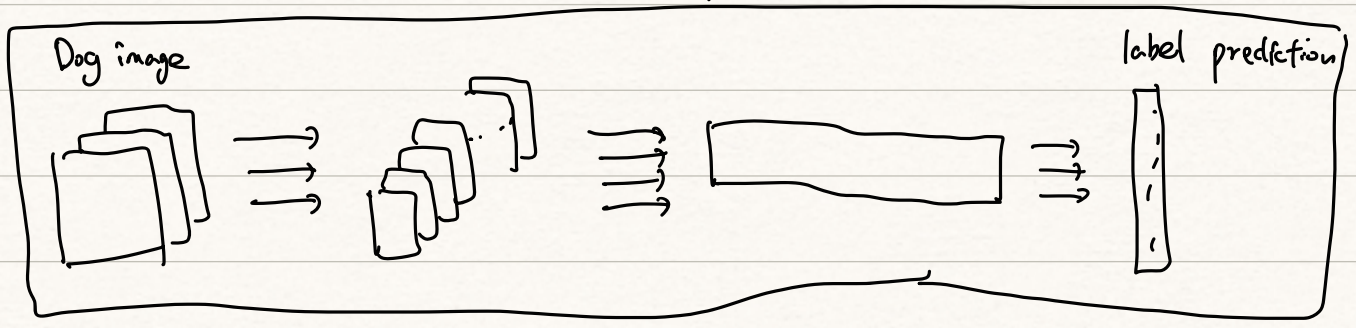
① 

Image Recognition CNN architecture (basis)

① $y_i = f^*(x_i)$ ⇝ choice 1: deterministic

$y_i = p^*(\cdot \mid x_i) \Leftrightarrow y_i = f^*(x_i) + \varepsilon_i$ ⇝ choice 2: random

⇑

Different Model

→ oracle f^* is unknown! we want to learn through ①

→ Approach

① Define Hypothesis class $\mathcal{H}$ ← specify ML model

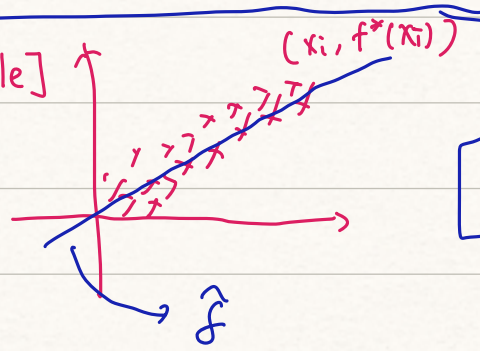
$\mathcal{H} = \{ f : f = w_0 + w_1 x \}$

↳ define a function class

↓

find a best f, namely $\hat{f}$ to approximate f^*

[Example]



- f^* → ground-truth mapping
- $\hat{f}$ → best we can achieve from $\mathcal{H}$

Rmk: from different $\mathcal{H}$ (hypothesis), we achieve different $\hat{f}$

(i.e. $\hat{f} \to \hat{f}_{\mathcal{H}}$)

Q: What is BEST APPROXIMATION means?

A: through LOSS FUNCTION!

[Example] $L(y', y) = \frac{1}{2}(y' - y)^2$

IDEA: Loss function: $L(y', y)$ { small if $y \approx y'$; large if $y \not\approx y'$

then BEST APPROXIMATION can be formulated as one optimization problem:

approximation of $\mathbb{E}_{x\sim\mathcal{D}}[L(f(x), f^*(x))]$

$$\min_{f\in\mathcal{H}} R_{emp}(f) = \min_{f\in\mathcal{H}} \frac{1}{N}\sum_{i=1}^{N} L(f(x_i), f^*(x_i))$$

- $f(x_i)$: output of our model
- $f^*(x_i)$: y_i (ground-truth label)
- $R_{emp}(f)$: average loss on the training set
- (one way) ERM Framework to achieve $\hat{f}$

$\Rightarrow$ convert learning approximation problem into optimization problem

$\Rightarrow$ use optimization framework to solve it

Recall: our goal is Generalization (on unseen data)

$$\hat{f} = \underset{f\in\mathcal{H}}{\operatorname{argmin}}\ R_{emp}(f)$$ $\longrightarrow$ not generalization

Approximation

Population risk minimization

Another choice $\rightarrow$ $\tilde{f} = \underset{f\in\mathcal{H}}{\operatorname{argmin}}\ R_{pop}(f)$

$= \underset{f\in\mathcal{H}}{\operatorname{argmin}}\ \mathbb{E}_{x\sim\mu}[L(f(x), f^*(x))]$

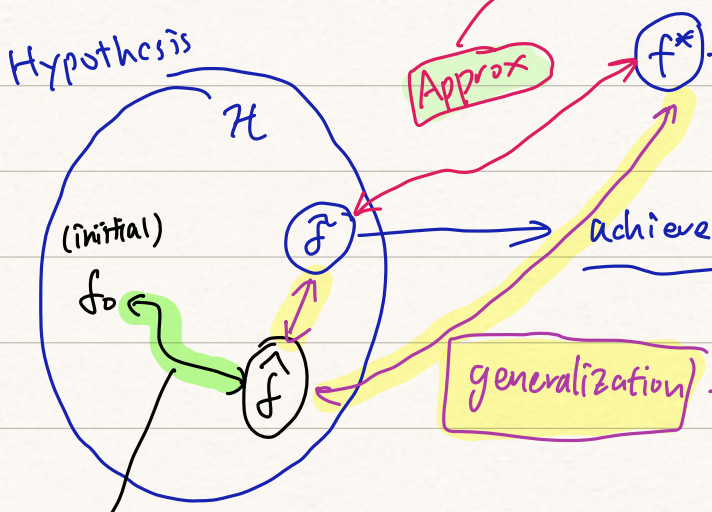
$= \underset{f\in\mathcal{H}}{\operatorname{argmin}}\ \mathbb{E}_{x\sim\mu}[L(f(x), y)]$

We want to solve

but cannot achieve

Summary

→ depends on how large our Hypothesis Space is



⇓

no guarantee for $\hat{f}$ to generalize well

⇓

Statistical ML

generalization : $\hat{f} \sim \tilde{f}$ or f^*

optimization : $f_0 \rightsquigarrow \hat{f}$

approximation : $\tilde{f} \sim f^*$

Problem → How to find a $\hat{f}$ to generalize well?

Linear Model

Rmk: $\min_{f \in \mathcal{H}} R_{emp}(f) = \min_{w \in \mathbb{R}^2} R_{emp}(w) = \min_{w \in \mathbb{R}^2} \frac{1}{2N} \|Y - Xw\|_2^2$

$X = \begin{pmatrix} 1 & x_1 \\ \vdots & \\ 1 & x_n \end{pmatrix}$

↳ Linear Regression

$\Rightarrow \mathcal{H} := \{ f : f = w_0 + w_1 x,\ w_0 \in \mathbb{R},\ w_1 \in \mathbb{R} \}$

$\Rightarrow L(y', y) = \frac{1}{2}(y' - y)^2$

$\Rightarrow \min_{f \in H} R_{emp}(f) = \min_{w_0, w_1} \frac{1}{2N} \sum_{i=1}^{N} (y_i - w_0 - w_1 x_i)^2$

↓

Necessary Condition $\begin{cases} \frac{\partial R_{emp}}{\partial w_0} = 0 \\ \frac{\partial R_{emp}}{\partial w_1} = 0 \end{cases} \rightarrow \begin{pmatrix} \hat{w}_0 \\ \hat{w}_1 \end{pmatrix}$

$\Rightarrow \hat{f}(x) = \hat{w}_0 + \hat{w}_1 x$

→ Question: Is $\mathcal{H}_{linear}$ good enough (complex enough)? → Small Hypo ⇒ under-fit

A: Depends on Task (Data). i.e. f^* → ground-truth mapping.

→ Overfitting Problem (capacity increases)

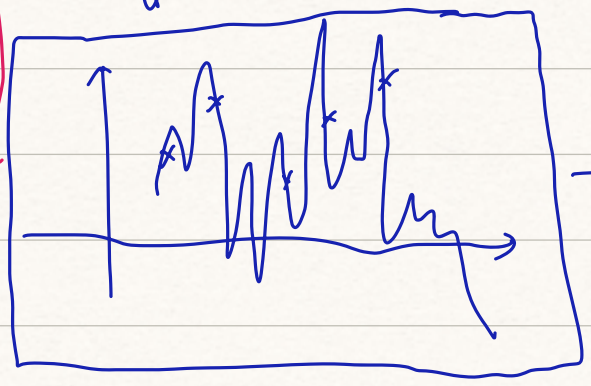
Consider $H_{99} = \{ f : f(x) = \sum_{i=0}^{99} w_i x^i \}$ ⇒ not linear model

↳ A larger hypothesis space

feature map

$\phi : x \longmapsto \begin{pmatrix} 1 \\ x \\ \vdots \\ x^{99} \end{pmatrix}$

heuristic result



hypothesis class { large overfit / small underfit

↑

overfitting problem

↳ { low loss in train / high loss in test

Linear Basis Model → Generalization of Linear Model

$\rightarrow H_M = \{ f : f(x) = \sum_{j=0}^{M-1} w_j \phi_j(x) \} = \{ f : f(x) = w^T \phi(x), w \in \mathbb{R}^M \}$

in which $\phi_j : \mathbb{R}^d \longrightarrow \mathbb{R}^M$ mapping (feature map)

$\Leftrightarrow$ Linear Model (on feature space) $\quad x \longrightarrow (\phi_0(x), \ldots, \phi_{M-1}(x))$

[By the Way, another generalization for Linear Regression is Neural Network]

(use activation function)

Learny Process

Model $\quad y_i = w^T \phi(x_i) = \phi^T(x_i) w$

$\phi(x_i) = \begin{pmatrix} \phi_0(x_i) \\ \vdots \\ \phi_{M-1}(x_i) \end{pmatrix}$

↓

$y = \Phi \cdot w \qquad \Phi = \begin{pmatrix} \phi(x_1)^T \\ \vdots \\ \phi(x_N)^T \end{pmatrix}$

↓

Loss $\min_{f \in \mathcal{H}_M} R_{emp}(f)$

$$= \min_{w \in \mathbb{R}^M} \frac{1}{2N} \| Y - \Phi w \|^2$$

$$= \min_{w} \frac{1}{2N} (Y - \Phi w)^T (Y - \Phi w)$$

$$= \min_{w} \frac{1}{2N} [\, Y^T Y - 2 w^T \Phi^T Y + w^T \Phi^T \Phi w \,]$$

$$\frac{\partial L}{\partial w} = \frac{1}{2N} [-2 Y^T \Phi + 2 w^T \Phi^T \Phi]$$

Necessary Condition

$$\nabla L = \frac{1}{2N} [-2 \Phi^T Y + 2 \Phi^T \Phi w] \equiv 0$$

$$\Rightarrow \hat{w} = (\Phi^T \Phi)^{-1} \Phi^T Y$$

actually is Sufficient

$\nabla^2 L = \Phi^T \Phi \geq 0$

(if $(\Phi^T \Phi)$ is invertible!!)

the feature map cannot be

So highly-dimensional

if not invertible, then Regularization

can be formulated as LM + Gaussian Noise

objective: $\min_w \frac{1}{2N} \| Y - \Phi w \|^2 + \lambda C(w)$

IDEA: Make w small

penalty term

$\lambda \rightarrow$ hyper-parameter

① $C(w) = \|w\|_2 \rightarrow$ Ridge regression (l2 reg)

② $C(w) = \|w\|_1$ → LASSO → advantage of sparse solution

(L_1 reg)

selection of features

Add PRIOR that, large weight will lead to over-fitting

Apply to K-class classification ← Softmax

(vector 2 probability distribution)

Linear Basis Model

→ $L(y', y)$ change to Cross Entropy Loss

$$:= -\sum_{k=1}^{K} y_k \log y'_k$$

$$= -y_{k(c)} \log y'_{k(c)}$$

$$= -\log y'_{k(c)}$$

$y \rightarrow$ ground-truth label

$y' \rightarrow$ output of model

Rmk: $L_{\text{cross-entropy}}(y, y') = 0 \iff y = y'$