Prob. Framework

"Machine Learning" is an algorithm that learns from experience w.r.t. some tasks and performance measure. i.e. its performance at given tasks will improve with experience under the perf. measure.

1) On the Set Ω. WE Ω is called Statistical Unit. We want to def a family $\mathcal{A}$ of "allowed 'yes-no'" questions. which's choosen as a σ-algebra

Rmk: It's logically consistent since trivial question $\Omega \in \mathcal{A}$; $A \in \mathcal{A}$.

$\Rightarrow$ negation question $A^c \in \mathcal{A}$. And we permit "and" connection.

2) Note that we can't be firecracker on which instance WE Ω to choose

We need to know the degree of the approvability $\mathbb{P}(A)$ for "yes" for $A \in \mathcal{A}$ so we introduce p.m. $\mathbb{P}$ on $\mathcal{A}$.

3') Often we don't distinguish instans w themselves. We introduce list of features $X(w)$ having value in $\mathbb{R}^1$. i.e. X is r.v. from Ω to $\mathbb{R}^n$.

$\Rightarrow$ We can assign prob. to events in feature space by:

$$X_* \mathbb{P}(B) := \mathbb{P}(X^{-1}(B)). \quad B \in \mathcal{B}_{\mathbb{R}^n}.$$

4) We also consider multiple acquisition of data: $\{X_j\}_{j \in \mathbb{N}}$ r.v.'s.

Remote: $X(n) = (X_1 \cdots X_n)$ sample of size n.

5') We'd like to get fresh info. from data acquisitions, which is realised by i.i.d. model (or (Stationary))

DTMC model $\{X_t\}_{t\in\mathbb{N}_0}$ with S (finite)

$$\mathbb{P}(X_t = s \mid X_0 = s_0, \dots, X_{t-1} = s_{t-1}) = \mathbb{P}(X_t = s \mid X_{t-1} = s_{t-1})$$

e.g. We may consider latent model $Z_j = (Y_j, X_j)$ with subcomponents Y_j as labels and X_j as covariates / features.

i) Design of Experiment (DoE):

Y_j is unpred. outcome and X_j is experiment cond'n controlled by us whose value is chosen deliberately to promote learning

ii) Transfer learning:

We deal with 2 sets of data $\{X_j\}$, $\{X_j'\}$, $X_j \overset{i.i.d.}{\sim} \mu$, $X_j' \overset{i.i.d.}{\sim} \mu'$.

where X_j is sparse / expensive. X_j' is abundant / cheap. And $\mu' \simeq \mu$.

$\Rightarrow$ We pre-train on X_j'. And tune it with limited experience X_j.