* Learning Problems

Computer Program is said to learn from experience E wrt T (class of tasks)
P (performance measure)

if its performance at task T improves with E

3 features of well defined problem

i) class of tasks (T)

ii) performance measure (P)

iii) experience (E)

Eg: Checkers

T → Playing checkers

P → % of games won

E → Playing practice games v/s itself

* Designing a Learning System

i) Choosing the training experience
- direct or indirect feedback
- degree to which learner controls sequence of eg
- how well it represents distribution of eg

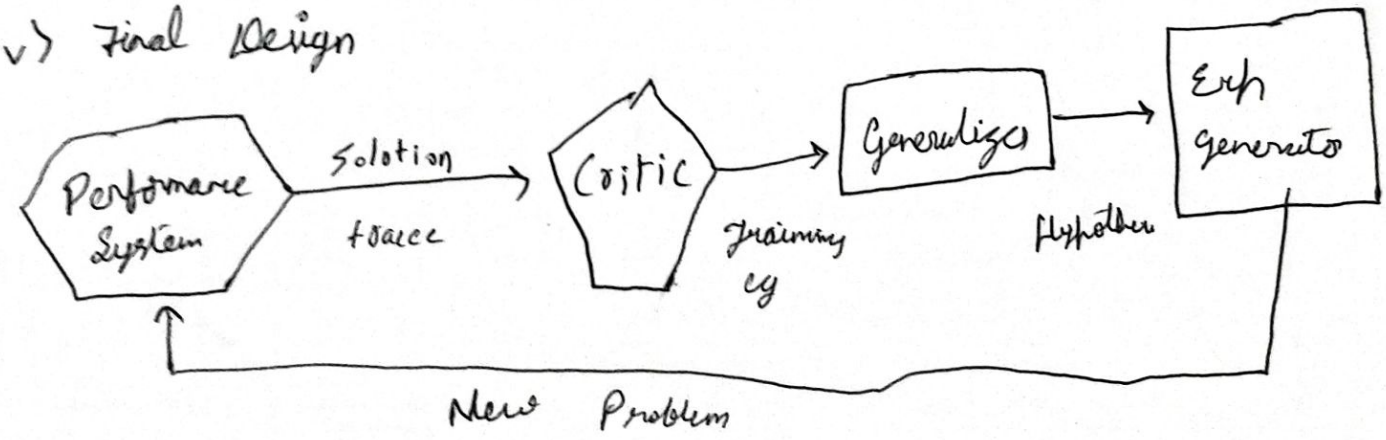
ii) Choosing the Target Function
- what type of knowledge is learned ?
- ChooseNextMove. (a) V s.t $V(b) = 100$ if b (board state) is W

iii) Choosing representation for target function
- linear comb repn
- $V(b) = w_1 x_1 + w_2 x_2 + \cdots + w_6 x_6$

$w \rightarrow$ weights $\quad x \rightarrow$ custom attributes

iv) Choosing function approximation algorithm
(- estimate training values, - adjust weights

v) Final Design



Perspective

Issues

CONCEPT LEARNING

"Inferring a boolean valued function from training examples of input and output."

Automatically inferring general def^n of some concept (bird) given examples labelled as members (or) non-members

Concept learning Task

Sky	Temp	Humidity	Wind	EnjoySport

"Learning set of days for which EnjoySport = YES"

Most general hypothesis : ? ? ? ? → YES for everything
Most specific hypothesis : ∅ ∅ ∅ ∅ → NO - -

X = Instances = Possible days

Attributes = S T H W

C = Target Concept = EnjoySport

H = Hypothesis = Conjunction of constraints on attributes
↳ (∅, ?, Warm)

DETERMINE h in H such that $h(x) = c(x)$

Concept Learning a Search

FIND - S

Finding maximally specific hypothesis

i) Initialize (h) to most specific hypothesis

ii) for each + instance x

for each attribute constraint (a) in h

if a is satisfied by x

do nothing

else

replace a in [h] by next more general ~~hypothesis~~ constraint

iii) Output h

In table 2.1

1) $h = \{\emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset\}$

2) After $x_1 = \{S, W, N, S, W, S\}$

$h = \{S, W, N, S, W, S\}$

3) After $x_2 = \{S, W, H, S, W, S\}$

$h = \{S, W, ?, S, W, S\}$

4) After $x_3 = \{\cdots\}$ {NO}

Ignore

5) After $x_4 = \{S, W, H, S, C, C\}$

$h = \{S, W, ?, S, ?, ?\}$

CANDIDATE ELIMINATION

Output set of all hypotheis consistent with training examples

* A hypotheis h is consistent with a set of training examples D iff $h(x) = c(x)$
~~for $\forall < x, c(x) >$ in D~~

* Version Space ($VS_{H,D}$) is subset of hypotheis from H consistent with training examples

Algorithm

Initialize G → maximally general h

Initialize S → maximally specific h

$S_0 = \{\emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset\}$

$S_1 = \{S, W, N, S, W, S\}$

$S_2 = \{S, W, ?, S, W, S\} = S_3$

$S_4 = \{S, W, ?, S, ?, ?\}$

$G_4 = \langle S, ????\rangle \langle ?W???\rangle$

$G_3 = \langle S, ????\rangle \langle ?, Warm, ????\rangle \langle ??N??\rangle \langle ???C?\rangle \langle ????Same\rangle$

$G_0 = \{?, ?, ?, ?, ?, ?\} = G_1 = G_2$

1) S_0 and G_0

2) S_1 and S_2 because target fn is YES & matches $G_0 = G_1 = G_2$

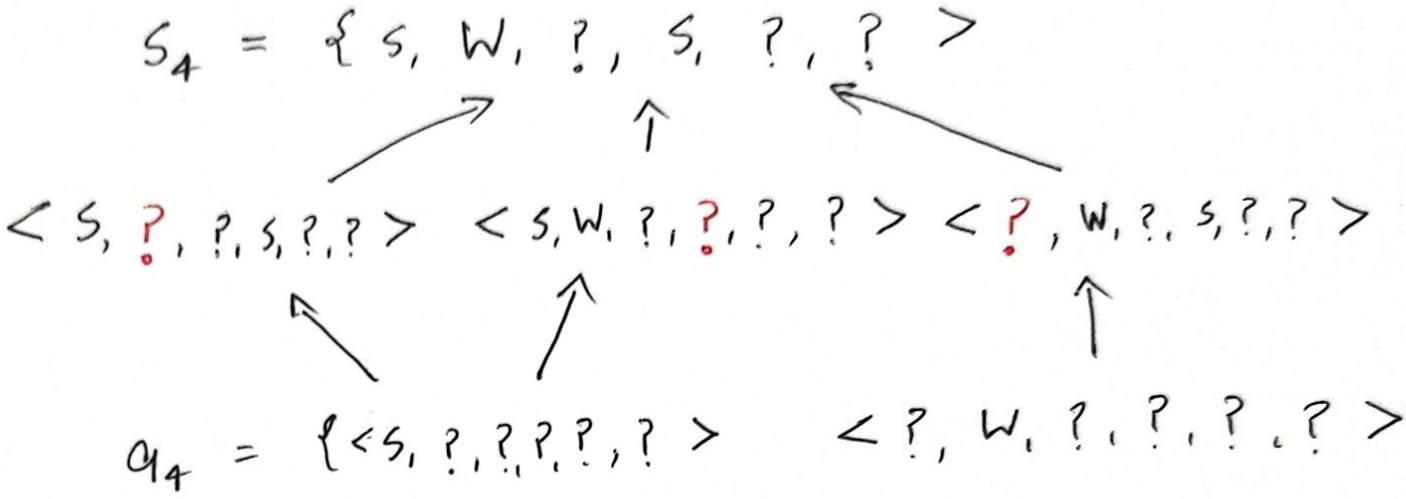
3) NO so $S_3 = S_2$

4) Opposite of example 3 is YES. Write combinations
   - Rainy → Sunny
   - Cold → Warm
   - High → Normal
   - Strong → nothing opp exists
   - ....

5) $\langle .. N .. \rangle$ and $\langle ... C .. \rangle$ is not consistent with S_3 so remove them

6) S_4 and also in G_4 $\langle ... S \rangle$ is inconsistent

Final Version Space (from S_4 and G_4)



---

* Inductive Bias