17 Hidden Markov Models and Partially Observable MDPs

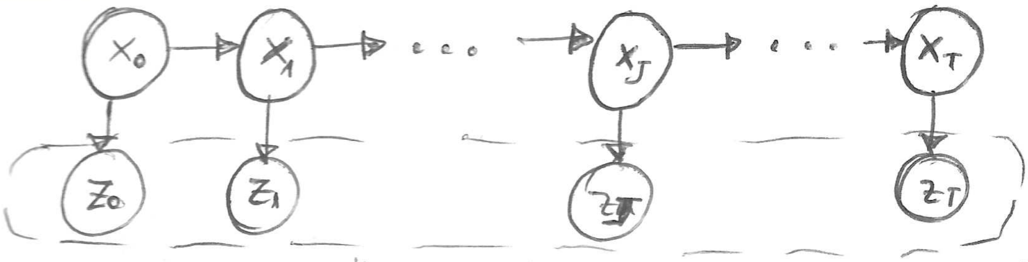
Markov Chain:

Dynamic system evolving according to the Markov property

$\underbrace{\text{FUTURE EVOLUTION DEPENDS ONLY ON THE CURRENT STATE}}_{\text{Markovian evolution.}}$

HIDDEN MARKOV MODELS

Useful to DESCRIBE SYSTEMS in which STATES are NOT FULLY OBSERVABLE. The agent cannot understand exactly the current config.



Also them follow the Markov property.

The agent can "observe" some configurations thanks to some sensors. In this model we don't have actions, since the agent cannot control the evolution of the system, THE INTEREST IS TO UNDERSTAND WHICH IS THE STATE GIVEN THE OBSERVATION.

$$HMM = \langle X, Z, \pi_0 \rangle$$

NO ACTIONS!

- transition model $P(x_t \mid x_{t-1})$
- observation model $P(z_t \mid x_t)$
- initial distribution

When we have finite and discrete states we can represent the probability distribution of the transitions in a matrix A, CALLED THE TRANSITION MATRIX:

$$A_{ij} = P(x_t = j \mid x_{t-1} = i)$$

The observation model can be either discrete or continuos. The initial probability is just $\pi_0 = P(x_0)$

Most of the recognition software (recognize the dynamic evolution of something) is an example of HMM.

In classification you have bunch of points without any order, HMM contains also the history and the position in the sequence is relevant (RECOGNIZE A WORD depends on letter position)

Most of the solutions are based on the Chain rule!

$$P(x_{0:T}, z_{1:T}) = P(x_0)P(z_0|x_0)P(x_1|x_0)P(z_1|x_1)P(x_2|x_1)\dots$$

You have to multiply **ALL ARROWS** in the model. There are two most important problem to solve:

1) $\boxed{\text{FILTERING}}$

estimation of current state given all observation recieved so far.

$$P(x_T = k | z_{1:T}) = \frac{\alpha_T^k}{\sum_J \alpha_T^J}$$

2) $\boxed{\text{SMOOTHING}}$

Estimate some past state given observation up to now.

$$P(x_t = k | z_{1:T}) = \frac{\alpha_t^k \beta_t^k}{\sum_J \alpha_t^J \beta_t^J}$$

$\boxed{t < T}$

"the past"

You can solve this two problems with the above formulas. Actual algorithms can be realized by computing α and β terms:

$$\boxed{\alpha_t \equiv P(x_t = k | z_{1:t})}$$

We can compute this quantity with forward step.

NOTE:

$z_{1:t}$
all observation from 1 to t

FORWARD STEP

- For each state k do:

$$\alpha_0^k = \pi_0 b_k(z_0)$$

observation model

- For each time $t = 1 \dots T$ do
  For each state k do:

$$\alpha_t^k = b_k(z_t) \sum_J \alpha_{t-1}^J A_{Jk}$$

While the β terms are the likelihood of observations:

$$\boxed{\beta_t^k = P(z_{t+1:T} \mid x_t = k)}$$

prediction of future observation given the current state.

BACKWARD STEP (starting from the final state)

- For each state k do:

$$\beta_T^k = 1$$

- For each time $t = T-1, \dots 1$ do
  For each state k

$$\beta_t^k = \sum_j \beta_{t+1}^J A_{kJ} b_J(z_{t+1})$$

Notice

FOR FILTERING β DOES NOT APPEAR! Coherent, we don't know anything of the future

What if we do not know the transition function and the observation model? In general we first estimate transition function and observation model, then we apply the above algorithms. How do we learn in HMM?

1 CASE: States can be observed at training time

You can do some experience and look to states, "a black box" that sometimes you can open. In this case you can easily estimate transition function and observation model with statistical analysis.

$$A_{iJ} = \frac{|\{i \to J \text{ transitions}\}|}{|\{i \to * \text{ transition}\}|}$$

$$b_k(v) = \frac{|\text{observe } v \wedge \text{state } k|}{|\text{observe } * \wedge \text{state } k|}$$

CASE 2: States cannot be observed, neither at training time.

Compute a local maximum likelihood with an EXPECTATION–MAXIMIZATION. Is possible to solve.

Recall the semplification of MDP and HMM:

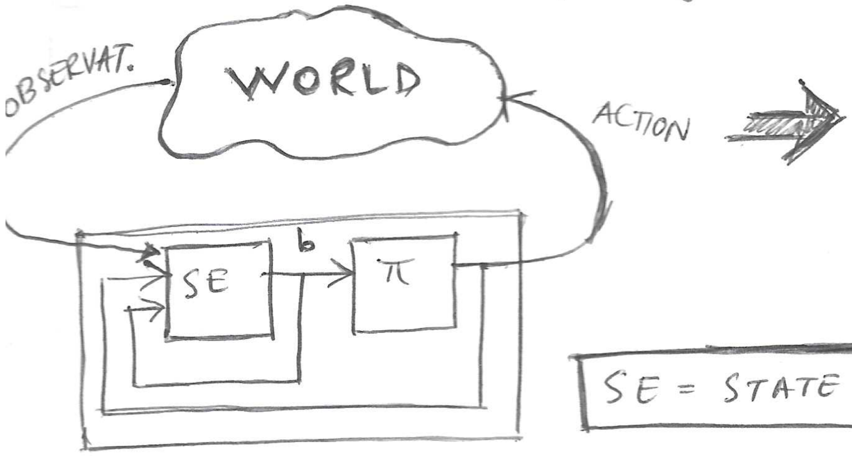
MDP $\Rightarrow$ states fully observable

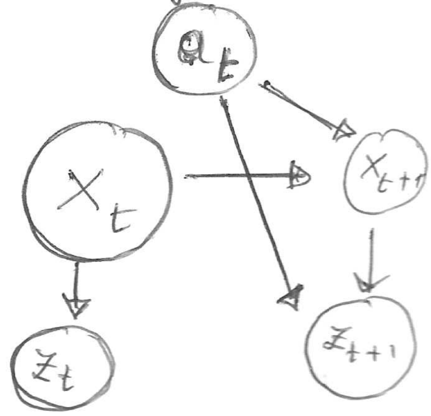
HMM $\Rightarrow$ avoid to control the system

} We can combine the system by considering!

POMDP = $\boxed{\text{PARTIALLY OBSERVABLE MDP}}$

The agent cannot observe directly the world but can receive some information, making then a process of STATE ESTIMATION in order to execute a policy.





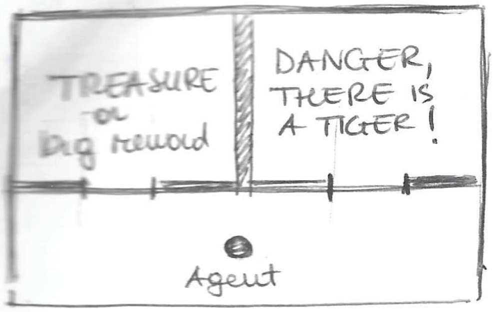
SE = STATE ESTIMATION component

POMDP = $\langle X, A, Z, \delta, r, o \rangle$

- X = states
- A = actions
- Z = observation
- $P(x_0)$ = prop. distr. of initial state.
- $\delta(x, a, x') = P(x' | x, a)$ is a prob distribution over transitions
- $r(x, a)$ = is a reward function
- $o(x', a, z') = P(z' | x', a)$ is a prob. distr. over observations.

combination of items coming from MDP and HMM

Example: Tiger problem



There is an agent in front of two doors, ~~both are~~ closed; the state of the world is the one in figure or the case in which the treasure and the tiger are switched.

$X = \{S_L, S_R\}$

$A = \{Open_L, Open_R, Listen\}$

$Z = \{t_R, t_L\}$ "t_i = tiger on i"

$P(x_0) = \langle 0.5, 0.5 \rangle$ The same probability of having the tiger on the left or on the right (no preferences)

$\delta(x, a, x')$. Listen does not change the state. Open actions are final action, after an open action the episode restarts.

$r(x,a)$ → +10 for opening the treasure door
→ −100 for " " tiger door
→ −1 if listening.

$o(x', a, z') = 0,85$ correct perception, 0.15 wrong perception

The observation function is defined only for the listen action. The observation model is the following:

	S_L	S_R
t_L	0.85	0,15
t_R	0.15	0.85

WHAT IS THE LISTEN ACTION?
WHAT DO YOU GAIN?
LISTEN IS an ACTION THAT INCREASE your KNOWLEDGE, increase the agent confidence!

POMDP → IS the <u>only model</u> in which you can introduce actions to gain knowledge (SENSING ACTIONS, knowledge introducing actions). These kind of actions are very important for a SMART agent.

Solution? The solution is still a policy but is not the same definition as in the MDP case, since the agent DOES NOT KNOW THE STATES. We have two options

Option 1: map from history of observations ~~to~~ actions

INPUT DOMAIN IS the set of all possible history of observations, VERY COMPLICATED FUNCTION

option 2: belief state

separate the state estimation phase and decision phase

$\boxed{\text{The belief state is an estimation of the current state}}$

THIS THE APPROACH WE FOLLOW.

Belief state $b(x)$ = probability distribution over states

POMDP can be described as an **MDP** in the belief states, but belief states are infinite:

- B is a set of belief states
- A is a set of actions
- $T(b,a,b')$ is a prob. distr. over transitions.
- $\rho(b,a,b')$ is a reward function

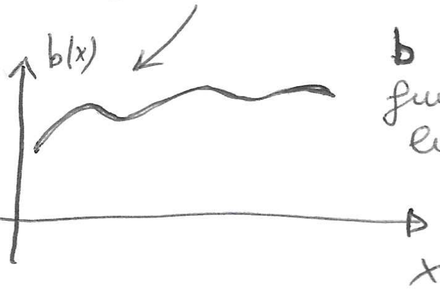
$$\boxed{\text{Policy } \pi : B \mapsto A}$$

the set of belief state

THERE EXIST THE SOLUTION.

$$b'(x') = SE(b,a,z') = P(x'|b,a,z')$$

$$\underbrace{= \frac{O(x',a,z') \sum_{x \in X} \delta(x,a,x') b(x)}{P(z'|b,a)}}$$



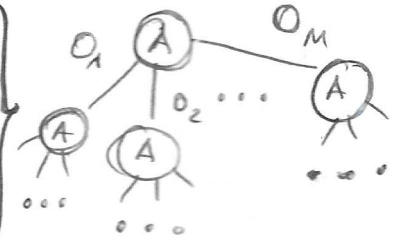
b is a function like this

given the current belief state, given the action I want to execute, and observations I receive, I can compute another belief state

. How can we solve the problem ? There are different methods. $b(x)$ can be difficult to approximate, but we can partion in intervals of states and for each interval the define a LINEAR FUNCTION. In this way for each interval we solve a LINEAR REGRESSION PROBLEM.

If we consider situations in which observations are discrete, we can introduce the <u>policy tree</u>: WE CAN REPRESENT THE HISTORY OF OBSERVATIONS IN A TREE.

Policy Tree:
- many levels
- in each level we have a choiche of an action and set of all possible observations in the branches.



(−) THE STRUCTURE IS HUGE, GROWS EXPONENTIONAL AT EVERY LEVEL.

let's go back to the tiger problem: (some idea of how it works)

We wanted to build the policy tree for this problem, the root can be one of three actions we have (Listen, $Open_R$, $Open_L$) and the ONE THAT WE CHOOSE IS THE ONE THAT MAXIMIZES THE VALUE FUNCTION.

For each policy it is possible to define a vector of all possible values of this policy for each possible states:

$$\alpha_\pi = \langle V^\pi(S_L), V^\pi(S_R) \rangle$$

In general α-vector is a value of n component, where in each component we put the value of that policy for each state.

π_1 : $Open_L$, π_2 : $Open_R$, π_3 : Listen ← One step policy

$$\alpha_{\pi_1} = \langle -100, 10 \rangle$$

$$\alpha_{\pi_2} = \langle 10, -100 \rangle$$

$$\alpha_{\pi_3} = \langle -1, -1 \rangle$$

Optimal one step policy: $V^{(1)}(b) = \max_{\pi} b\, \alpha_{\pi}$
These are the expected values of α-vectors:

$$\alpha_1 = 0.5(-100) + 0.5(10) = -45$$

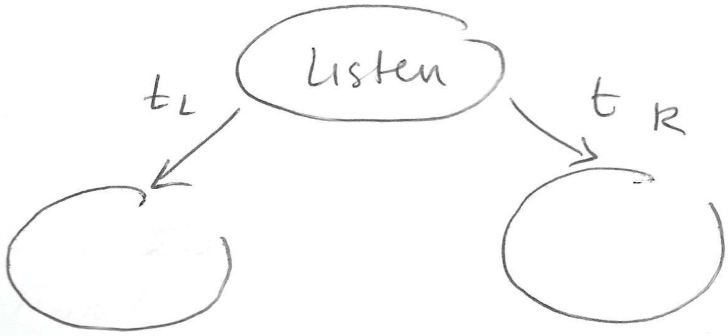
$$\alpha_2 = 0.5(10) + 0.5(-100) = -45$$

$$\alpha_3 = 0.5(-1) + 0.5(-1) = -1$$

$\leftarrow$ best policy

This depends of the numbers

One possible way of applying this approach in a greedy way. In principle we should expand each policy tree, but in any step we take the best policy, discard all the others, and build the other step policy by the one selected. In the previous case we should select α_3 and discard the others!



From α_3 I can build the two step policy.

Let' see some two step policies!

π_1 = Listen; (t_L: Listen, t_R: Listen) $\rightarrow \alpha_{\pi_1} = \langle -2, -2 \rangle$

π_2 = $\underbrace{\text{Listen; } (t_L : \text{Open}_R \text{ ; } t_R : \text{Open}_L)}_{\text{depending whether I observe the tiger I decide to open the opposite-side-door}} = \alpha_{\pi_2} = \langle ?, ? \rangle$

$\Rightarrow$ seems reasonable

If we are in S_L we will have the transition with prob. 0.85 the other with 0.15.

$$V^{\pi}(S_L) = -1 + 0.85(+10) + 0.15(-100) =$$
$$= -1 + 8{,}5 - 15 = -7{,}5$$

$$V^{\pi}(S_R) = -1 + 0.85(+10) + 0{,}15(-100) = -7{,}5$$

$\alpha = \langle -7{,}5, -7{,}5 \rangle$