Sapienza University of Rome

Master in Artificial Intelligence and Robotics
Master in Engineering in Computer Science

Machine Learning

A.Y. 2020/2021

Prof. L. Iocchi, F. Patrizi

19. Hidden Markov Models and Partially Observable MDPs

L. Iocchi, F. Patrizi

Overview

- Hidden Markov Models (HMM)
- Learning in HMM
- Partially Observable Markov Decision Processes (POMDP)
- Policy trees
- Example: POMDP tiger proglem

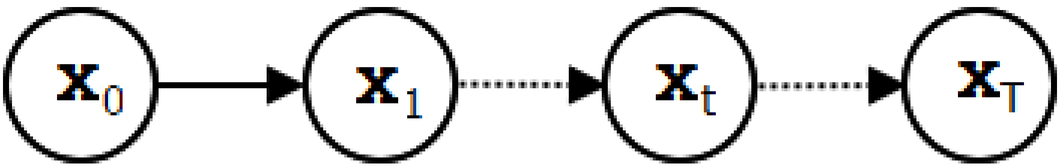
Markov Chain

each state depends only on the previous and not from the history

something that evolves overtime

Dynamic system evolving according to the Markov property.

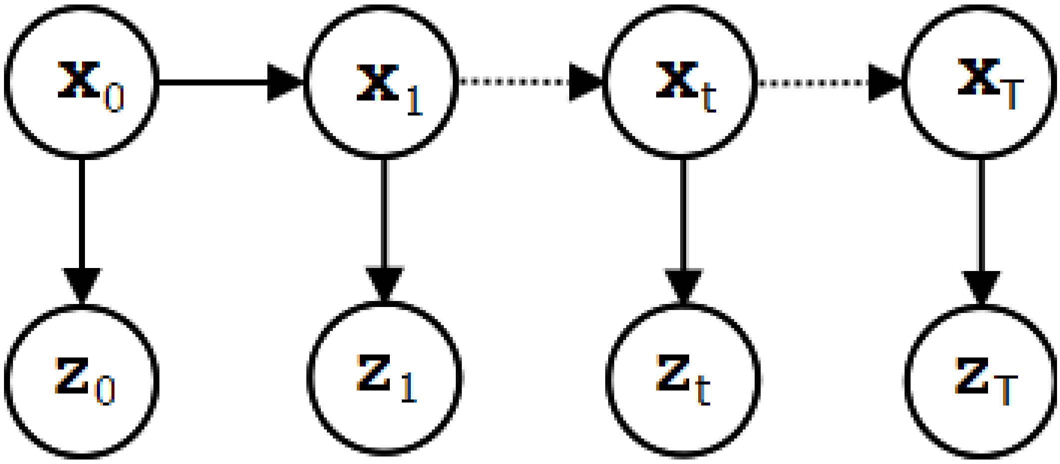
dynamic BN that models a random process in which state variables are defined in term of random variables



Future evolution depends only on the current state $\mathbf{X}_t$

Hidden Markov Models (HMM)

states that are hidde and so no directly observable



random variables called observations on which we can observe the value

- states $\mathbf{x}_t$ are **discrete** and **non-observable**,
- observations (emissions) $\mathbf{z}_t$ can be either discrete or continuous.
- controls $\mathbf{u}_t$ are not present (i.e., evolution is not controlled by our system),

HMM representation

formal definition of Hidden Markov Model

POSSIBLE EXAM QUESTION:
write down a difference between MDP and HMM. this slide is the perfect answer on exam question

$$\text{HMM} = \langle \mathbf{X}, \mathbf{Z}, \pi_0 \rangle$$

state transition probability

- transition model: $P(\mathbf{x}_t | \mathbf{x}_{t-1})$
- observation model: $P(\mathbf{z}_t | \mathbf{x}_t)$
- initial distribution: π_0

initial probability distribution

model with a certain number of states and a set of observations

2 functions that describe the transitions one for the states and one for the observations

State transition matrix $\mathbf{A} = \{A_{ij}\}$

$$P(x_t = \{a,b,c\} \,|\, x_{t-1} = \{a,b,c\})$$

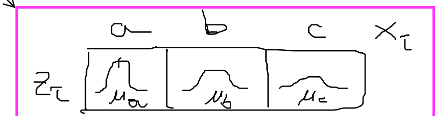
since these values are discrete this probability distribution can be express in form of a matrix

$x_t \backslash x_{t-1}$	a	b	c
a		0.3	
b		0.5	
c			

$$A_{ij} \equiv P(\mathbf{x}_t = j | \mathbf{x}_{t-1} = i)$$

Observation model (discrete or continuous):

$$b_k(\mathbf{z}_t) \equiv P(\mathbf{z}_t | \mathbf{x}_t = k)$$



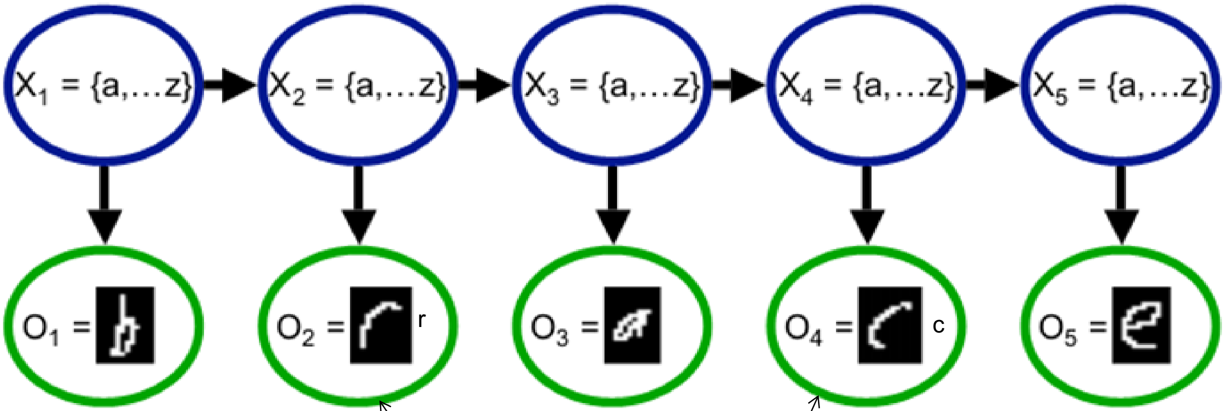
in case of continuous model like a gaussian probability distribution

Initial probabilities:

$$\pi_0 = P(\mathbf{x}_0)$$

HMM examples of applications

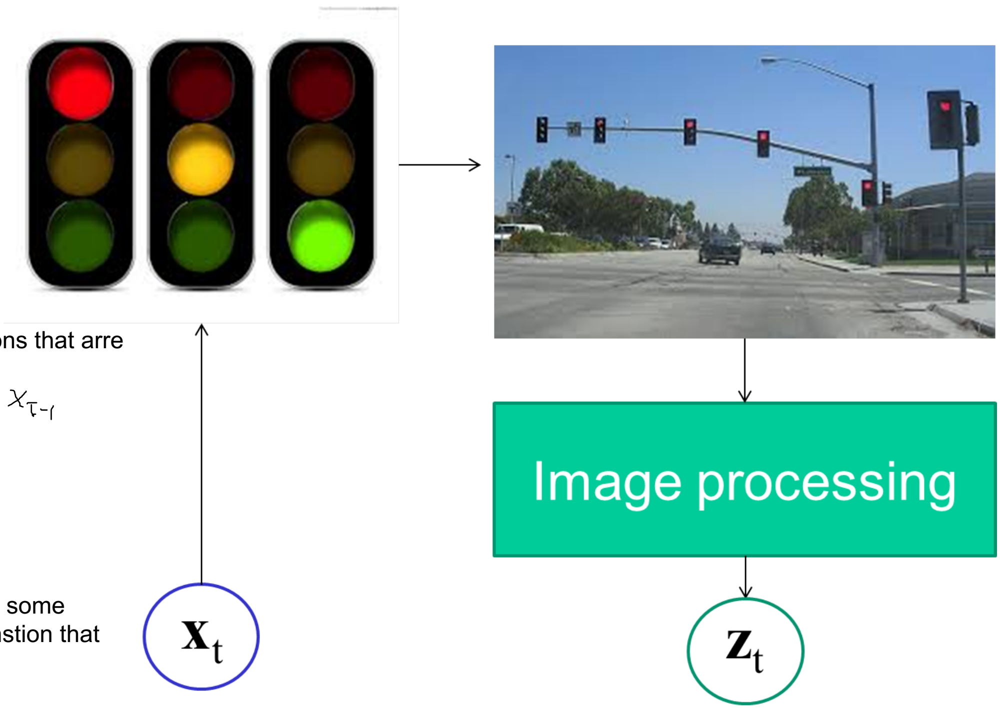
Handwriting recognition



if we remove the context is more difficult to understand which is r and which is c

Similar structure for speech/gesture/activity recognition.

HMM examples of applications



there are some transitions that arre not allowed

	R	Y	G x_{t-1}
R			0
x_t Y			
G	0		

the markov model adds some knowledge, like the transtion that are not allowed

HMM factorization

Application of chain rule on HMM:

$$P(\mathbf{x}_{0:T}, \mathbf{z}_{1:T}) = P(\mathbf{x}_0)P(\mathbf{z}_0|\mathbf{x}_0)P(\mathbf{x}_1|\mathbf{x}_0)P(\mathbf{z}_1|\mathbf{x}_1)P(\mathbf{x}_2|\mathbf{x}_1)P(\mathbf{z}_2|\mathbf{x}_2)\dots$$

HMM inference

Given HMM $= \langle \mathbf{X}, \mathbf{Z}, \pi_0 \rangle$,

Filtering problem of estimating the state of the system at time t given all the past observations

$$P(\mathbf{x}_T = k|\mathbf{z}_{1:T}) = \frac{\alpha_T^k}{\sum_j \alpha_T^j}$$

Smoothing

$$P(\mathbf{x}_t = k|\mathbf{z}_{1:T}) = \frac{\alpha_t^k \beta_t^k}{\sum_j \alpha_t^j \beta_t^j}$$

Forward step

computed in Alpha term

Forward iterative steps to compute

$$\alpha_t^k \equiv P(\mathbf{x}_t = k, \mathbf{z}_{1:t})$$

- For each state k do:
  - $\alpha_0^k = \pi_0 b_k(\mathbf{z}_0)$
- For each time $t = 1, \ldots, T$ do:
  - For each state k do:
    - $\alpha_t^k = b_k(\mathbf{z}_t) \sum_j \alpha_{t-1}^j A_{jk}$

Backward step

beta is used

Backward iterative steps to compute

$$\beta_t^k \equiv P(\mathbf{z}_{t+1:T} | \mathbf{x}_t = k)$$

- For each state k do:
  - $\beta_T^k = 1$
- For each time $t = T-1, \ldots, 1$ do:
  - For each state k do:
    - $\beta_t^k = \sum_j \beta_{t+1}^j \underbrace{A_{kj}}_{\text{transition matrix}} \underbrace{b_j(\mathbf{z}_{t+1})}_{\text{observation function}}$

Learning in HMM

until now we have assumed that the info about the model are given, but here we will se what happens in the case in which we don't know the transition model and the observation model. We have 2 cases

Given output sequences, determine maximum likelihood estimate of the parameters of the HMM (*transition and emission probabilities*).

Case 1: states can be observed at training time

there is a moment in which the system is open and we can look at the states

Transition and observation models can be estimated with statistical analysis

$$A_{ij} = \frac{|\{i \rightarrow j \ transitions\}|}{|\{i \rightarrow * \ transitions\}|}$$

from i to any other astate

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

total number of events in which I observe anything from state k

Sapienza University of Rome, Italy - Machine Learning (2020/2021)

Learning in HMM

we can solve in a similar way for unsupervised learning

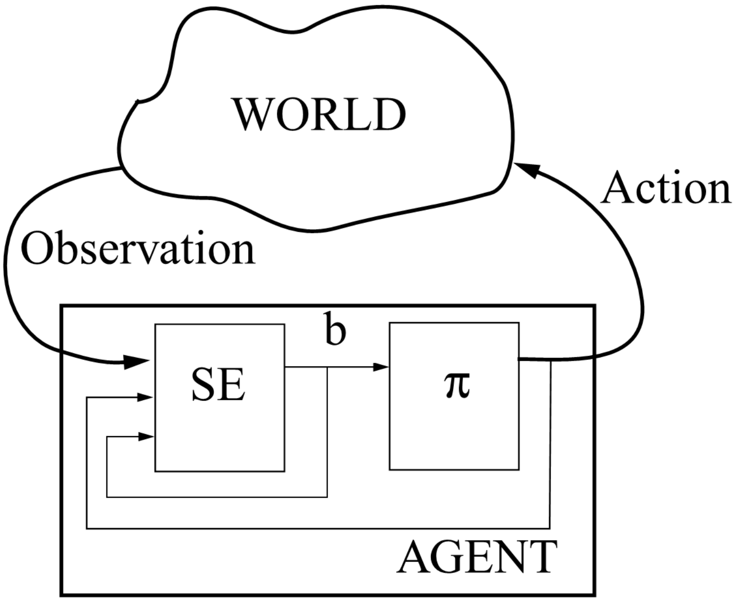
Case 2: states cannot be observed at training time

the states are always hidden

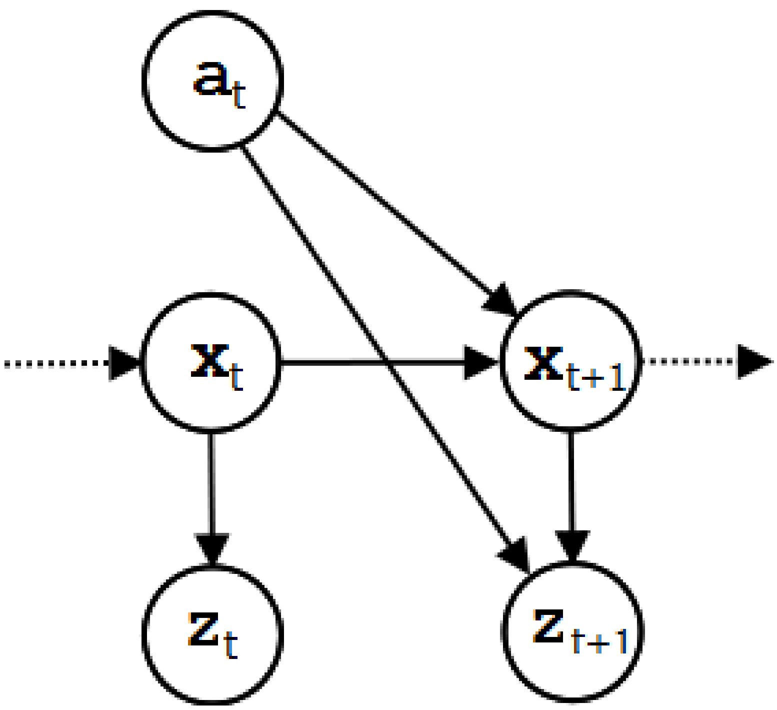
Compute a **local** maximum likelihood with an Expectation-Maximization (EM) method (e.g., Baum-Welch algorithm).

POMDP agent

Combines decision making of MDP and non-observability of HMM.



POMDP graphical model



POMDP representation

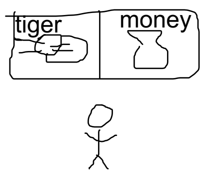
union of all the lements that we have seen in MDPs and HMM

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

- $\mathbf{X}$ is a set of states
- $\mathbf{A}$ is a set of actions
- $\mathbf{Z}$ is a set of observations
- $P(\mathbf{x}_0)$ is a probability distribution of the initial state
- transition function
- $\delta(\mathbf{x}, a, \mathbf{x}') = P(\mathbf{x}'|\mathbf{x}, a)$ is a probability distribution over transitions
- $r(\mathbf{x}, a)$ is a reward function
- $o(\mathbf{x}', a, \mathbf{z}') = P(\mathbf{z}'|\mathbf{x}', a)$ is a probability distribution over observations

observation function

Example: tiger problem



env with two rooms we are outside the room and we know that in one room there is a tiger and that in the other there is a trasure. We have 2 possible state (when the tiger is on the left and when it is on the right)

Two closed doors hide a treasure and a tiger.

$X = \{s_L, s_R\}$

situation in which the tiger is on the left and situation in which the tiger is on the right. But we don't know in which situations we are

- $\mathbf{X} = \{s_L, s_R\}$
- open left door open right door
- $\mathbf{A} = \{Open_L, Open_R, Listen\}$ so we have 3 actions
- $\mathbf{Z} = \{t_L, t_R\}$ we observe the tiger to be on the left, to be on the right
- $P(\mathbf{x}_0) =< 0.5, 0.5 >$ initial probability at the beginning
- $\delta(\mathbf{x}, a, \cancel{\mathbf{x}'})$ $Listen$ does not change state, $Open$ actions restart the situation with 0.5 probability between s_L, s_R
- $r(\mathbf{x}, a) = 10$ if opening the treasure door, -100 if opening the tiger door, -1 if listening
- $o(\mathbf{x}', a, \mathbf{z}') = 0.85$ correct perception, 0.15 wrong perception

GO TO SLIDE 25

Solution concept for POMDP

Solution: *policy*, but we do not know the states!

Option 1: map from history of observations to actions
- histories are too long!

Option 2: belief state the belief (so the estimate at time t) can be used to choose an action
- probability distribution over the current state

Belief MDP

Belief $b(\mathbf{x})$ = probability distribution over the states.

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

- $\mathbf{B}$ is a set of belief states
- $\mathbf{A}$ is a set of actions
- $\tau(b, a, b')$ is a probability distribution over transitions
- $\rho(b, a, b')$ is a reward function

Policy: $\pi : \mathbf{B} \mapsto \mathbf{A}$

Computing Belief States

Given current belief state b, action a and observation $\mathbf{z}'$ observed after execution of a, compute the next belief state $b'(\mathbf{x}')$

$$
\begin{aligned}
b'(\mathbf{x}') &\equiv SE(b, a, \mathbf{z}') \equiv P(\mathbf{x}'|b, a, \mathbf{z}') \\
&= \frac{P(\mathbf{z}'|\mathbf{x}', b, a)P(\mathbf{x}'|b, a)}{P(\mathbf{z}'|b, a)} \\
&= \frac{P(\mathbf{z}'|\mathbf{x}', a)\sum_{\mathbf{x}\in\mathbf{X}} P(\mathbf{x}'|b, a, \mathbf{x})P(\mathbf{x}|b, a)}{P(\mathbf{z}'|b, a)} \\
&= \frac{o(\mathbf{x}', a, \mathbf{z}')\sum_{\mathbf{x}\in\mathbf{X}} \delta(\mathbf{x}, a, \mathbf{x}')b(\mathbf{x})}{P(\mathbf{z}'|b, a)}
\end{aligned}
$$

Belief MDP transition and reward functions

Transition function

$$\tau(b, a, b') = P(b'|b, a) = \sum_{\mathbf{z}\in\mathbf{Z}} P(b'|b, a, \mathbf{z})P(\mathbf{z}|b, a)$$

$$P(b'|b, a, \mathbf{z}) = 1 if\ b' = SE(a, b, \mathbf{z}),\ 0\ otherwise$$

Reward function

$$\rho(b, a) = \sum_{\mathbf{x}\in\mathbf{X}} b(\mathbf{x})r(\mathbf{x}, a)$$

Value function in POMDP

$$V(b) = \max_{a \in \mathbf{A}} [\rho(b,a) + \gamma \sum_{b'} (\tau(b,a,b') V(b'))]$$

Replacing $\tau(b,a,b')$ and $\rho(b,a)$ and considering that $P(b'|b,a,\mathbf{z}) = 1$, if $b' = SE(a,b,\mathbf{z}) = b^a_{\mathbf{z}}$, and 0 otherwise

$$V(b) = \max_{a \in \mathbf{A}} [\sum_{\mathbf{x} \in \mathbf{X}} b(\mathbf{x}) r(\mathbf{x},a) + \gamma \sum_{\mathbf{z} \in \mathbf{Z}} P(\mathbf{z}|b,a) V(b^a_{\mathbf{z}})]$$

Value iteration for belief MDP

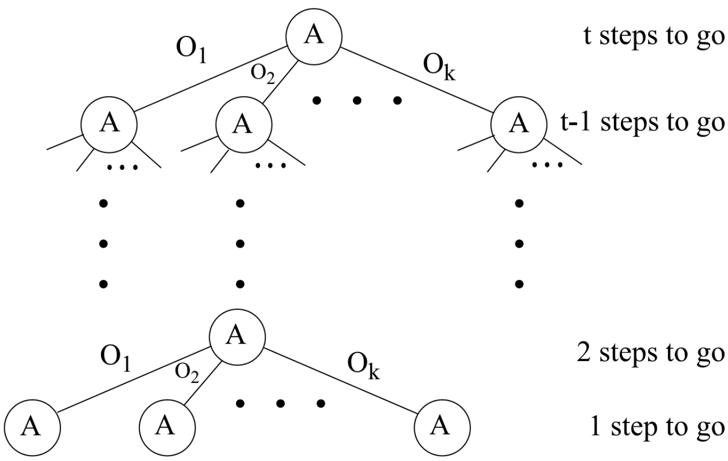
- Discretize the distributions $b(\mathbf{x})$
- Apply value iteration on the discretized belief MDP

A similar method can be devised for any MDP solving technique.

Solution concept in POMDP

Policy trees

representation of a possible solution concept in which what you do is, given the initial knowledge, we choose an action, we execute the action and then we get observation and depending on the observation we get we choose the next action. then we receive observation...and so on...



Value function for tiger problem

3 policies

One-step policies: $\pi_1 = Open_L$, $\pi_2 = Open_R$, $\pi_3 = Listen$

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

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

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

One-step optimal value function:

$$V^{(1)}(b) = \max_{\pi} b\, \alpha_{\pi}$$

$V_{\pi_1} = -100\ 0.5 + 10\ 0.5 = -45$

$V_{\pi_2} = \quad = -45$

$V_{\pi_3} = -1$

Value function for tiger problem

$$\begin{array}{ccc} & \text{Listen} \;\; \pi_1 & \\ {\scriptstyle t_L} \swarrow & & \searrow {\scriptstyle t_R} \\ \text{Listen} & & \text{Listen} \end{array}$$

Two-step policies:

$\pi_1 = Listen; (t_L : Listen, t_R : Listen) \rightarrow \alpha_{\pi_1} = \langle -2, -2 \rangle$
$\pi_2 = Listen; (t_L : Open_R, t_R : Open_L) \rightarrow \alpha_{\pi_2} = \langle -7.5, -7.5 \rangle$
$\pi_3 = Open_L; (t_L : Open_L, t_R : Open_L) \rightarrow \alpha_{\pi_3} = \langle -145, -35 \rangle$
$\pi_4 = Open_L; (t_L : Listen, t_R : Listen) \rightarrow \alpha_{\pi_4} = \langle -101, 9 \rangle$
$\pi_5 = Open_R; (t_L : Listen, t_R : Listen) \rightarrow \alpha_{\pi_5} = \langle 9, -101 \rangle$
... and many others

Two-step optimal value function:

$$V^{(2)}(b) = \max_{\pi} b\, \alpha_{\pi}$$

Value function for tiger problem

Three-step policies:

$\pi_1 = Listen; Listen; (t_L, t_L : Open_R, t_R, t_R : Open_L, t_L, t_R \text{ or } t_R, t_L : Listen)$
... and many many others ...

Three-step optimal value function:

$$V^{(3)}(b) = \max_{\pi} b\, \alpha_{\pi}$$

References

Leslie Pack Kaelbling, Michael L. Littman, Anthony R. Cassandra.
Planning and acting in partially observable stochastic domains.
Artificial Intelligence, vol. 101, issues 12, 1998, pages 99134.