Sapienza University of Rome

Master in Artificial Intelligence and Robotics

Machine Learning

A.Y. 2025/2026

Prof. Luca Iocchi

Sapienza University of Rome, Italy - Machine Learning (2024/2025)

19. Hidden Markov Models and Partially Observable MDPs

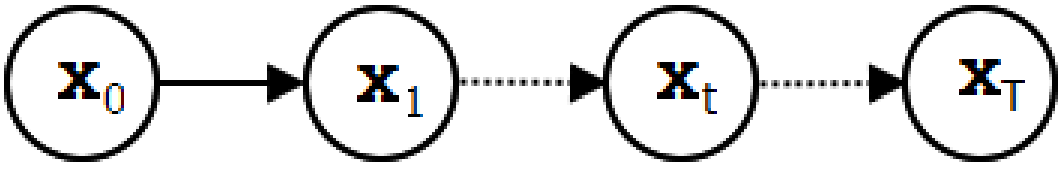
Luca Iocchi

Overview

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

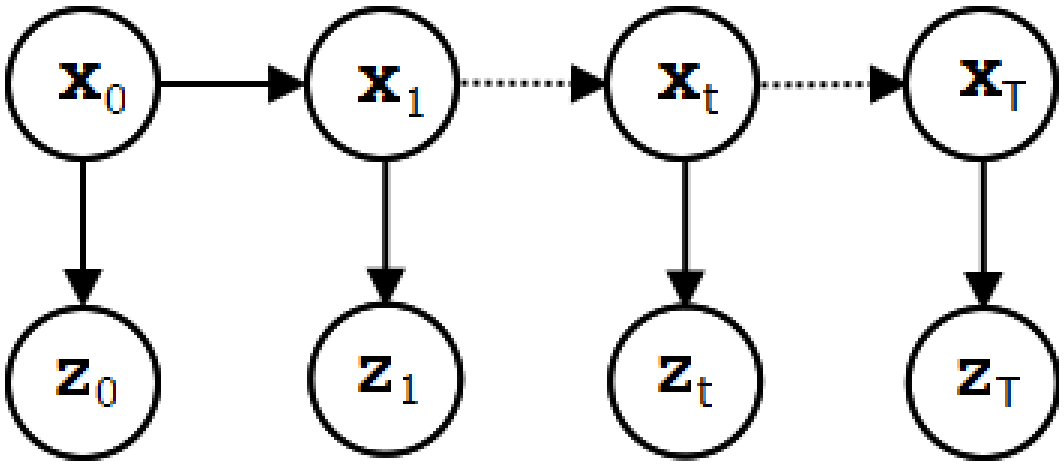
Markov Chain

Dynamic system evolving according to the Markov property.



Future evolution depends only on the current state X_t

Hidden Markov Models (HMM)



- states x_t are **discrete** and **non-observable**,
- observations (emissions) z_t can be either discrete or continuous.
- controls u_t are not present (i.e., evolution is not controlled by our system),

HMM representation

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

- transition model: $P(x_t|x_{t-1})$
- observation model: $P(z_t|x_t)$
- initial distribution: π_0

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

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

Observation model (discrete or continuous):

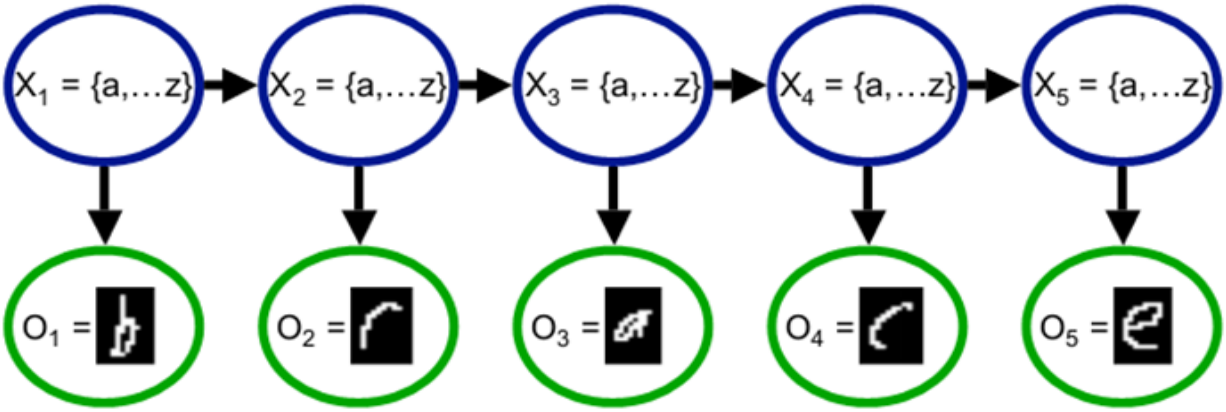
$$b_k(z_t) \equiv P(z_t|x_t = k)$$

Initial probabilities:

$$\pi_0 = P(x_0)$$

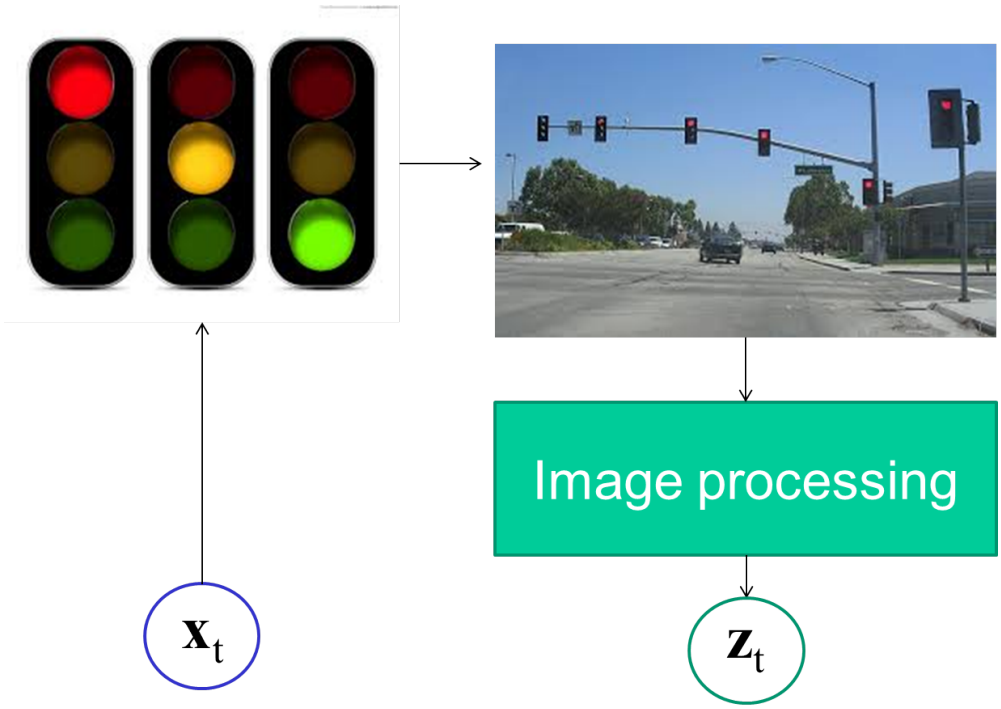
HMM examples of applications

Handwriting recognition



Similar structure for speech/gesture/activity recognition.

HMM examples of applications



HMM factorization

Application of chain rule on HMM:

$$P(x_{0:T}, z_{1:T}) = P(x_0)P(z_0|x_0)P(x_1|x_0)P(z_1|x_1)P(x_2|x_1)P(z_2|x_2)\dots$$

HMM inference

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

Filtering

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

Smoothing

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

Forward step

Forward iterative steps to compute

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

- For each state k do:
  - $\alpha_0^k = \pi_0 b_k(z_0)$
- For each time $t = 1, \ldots, T$ do:
  - For each state k do:
    - $\alpha_t^k = b_k(z_t) \sum_j \alpha_{t-1}^j A_{jk}$

Backward step

Backward iterative steps to compute

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

- For each state k do:
  - $\beta_T^k = 1$
- For each time $t = T - 1, \ldots, 1$ do:
  - For each state k do:
    - $\beta_t^k = \sum_j \beta_{t+1}^j A_{kj} b_j(z_{t+1})$

Learning in HMM

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

Transition and observation models can be estimated with statistical analysis

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

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

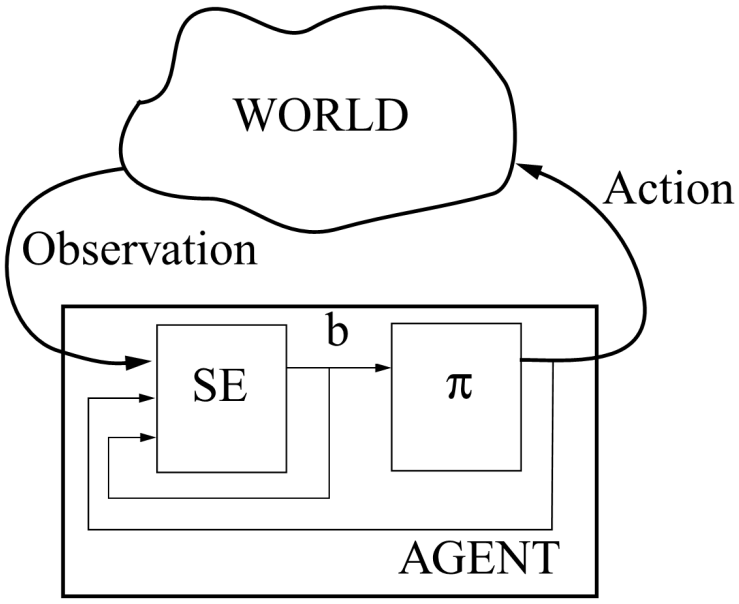
Learning in HMM

Case 2: states cannot be observed at training time

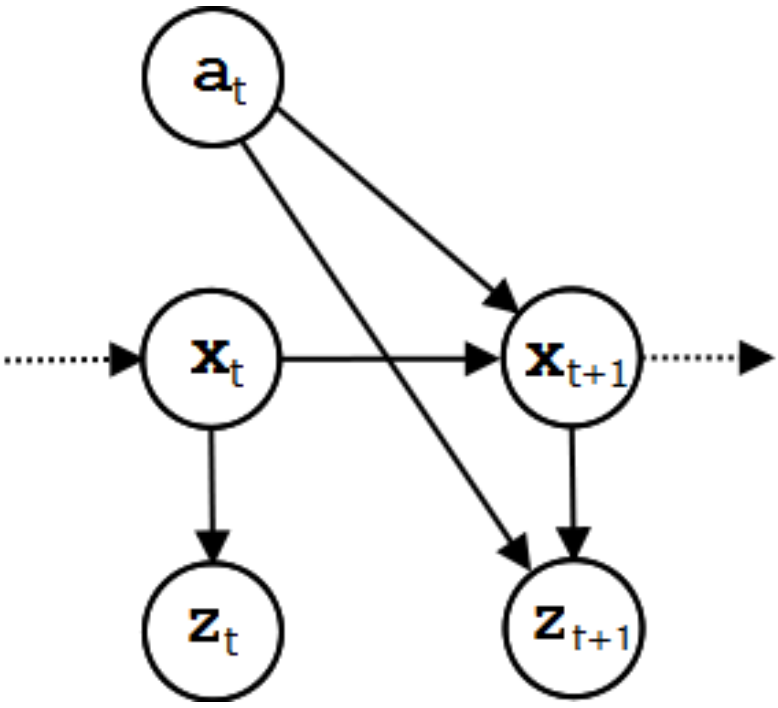
Compute a **local** maximum likelihood with an Expectation-Maximization (EM) method (e.g., Baum-Welch algorithm).

POMDP agent

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



POMDP graphical model



POMDP representation

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

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

Example: tiger problem

Two closed doors hide a treasure and a tiger.

- $\mathrm{X} = \{s_L, s_R\}$
- $\mathrm{A} = \{Open_L, Open_R, Listen\}$
- $\mathrm{Z} = \{t_L, t_R\}$
- $P(\mathrm{x}_0) =< 0.5, 0.5 >$
- $\delta(\mathrm{x}, a, \mathrm{x}')$ *Listen* does not change state, *Open* actions restart the situation with 0.5 probability between s_L, s_R
- $r(\mathrm{x}, a) = 10$ if opening the treasure door, -100 if opening the tiger door, -1 if listening
- $o(\mathrm{x}', a, \mathrm{z}') = 0.85$ correct perception, 0.15 wrong perception

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
- probability distribution over the current state

Belief MDP

Belief $b(x)$ = probability distribution over the 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
- $\tau(b, a, b')$ is a probability distribution over transitions
- $\rho(b, a, b')$ is a reward function

Policy: $\pi : B \mapsto A$

Computing Belief States

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

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

Belief MDP transition and reward functions

Transition function

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

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

Reward function

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

Value function in POMDP

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

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

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

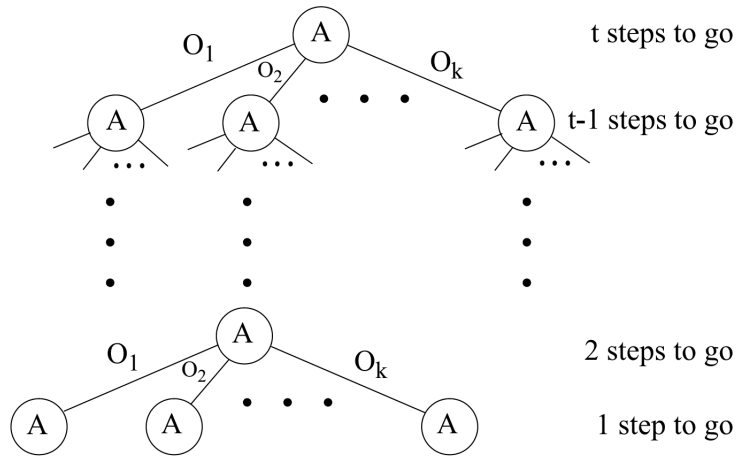
Value iteration for belief MDP

- Discretize the distributions $b(x)$
- 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



Value function for tiger problem

One-step policies: $\pi_1 = \mathit{Open}_L$, $\pi_2 = \mathit{Open}_R$, $\pi_3 = \mathit{Listen}$

$$\alpha_{\pi_1} = \langle -100, 10 \rangle$$
$$\alpha_{\pi_2} = \langle 10, -100 \rangle$$
$$\alpha_{\pi_3} = \langle -1, -1 \rangle$$

One-step optimal value function:

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

Value function for tiger problem

Two-step policies:
$\pi_1 = \mathit{Listen}; (t_L : \mathit{Listen}, t_R : \mathit{Listen}) \rightarrow \alpha_{\pi_1} = \langle -2, -2 \rangle$
$\pi_2 = \mathit{Listen}; (t_L : \mathit{Open}_R, t_R : \mathit{Open}_L) \rightarrow \alpha_{\pi_2} = \langle -7.5, -7.5 \rangle$
$\pi_3 = \mathit{Open}_L; (t_L : \mathit{Open}_L, t_R : \mathit{Open}_L) \rightarrow \alpha_{\pi_3} = \langle -145, -35 \rangle$
$\pi_4 = \mathit{Open}_L; (t_L : \mathit{Listen}, t_R : \mathit{Listen}) \rightarrow \alpha_{\pi_4} = \langle -101, 9 \rangle$
$\pi_5 = \mathit{Open}_R; (t_L : \mathit{Listen}, t_R : \mathit{Listen}) \rightarrow \alpha_{\pi_5} = \langle 9, -101 \rangle$
... and many others

Two-step optimal value function:

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

Value function for tiger problem

Three-step policies:
$\pi_1 = \mathit{Listen}; \mathit{Listen}; (t_L, t_L : \mathit{Open}_R, t_R, t_R : \mathit{Open}_L, t_L, t_R \text{ or } t_R, t_L : \mathit{Listen})$
... 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 1–2, 1998, pages 99–134.