CSE 150A-250A AI: Probabilistic Models

Lecture 14

Fall 2025

Trevor Bonjour
Department of Computer Science and Engineering
University of California, San Diego

Slides adapted from previous versions of the course (Prof. Lawrence, Prof. Alvarado, Prof Berg-Kirkpatrick)

Agenda

Review

Reinforcement Learning

Review

EM algorithm for HMMs

- CPTs to re-estimate:

$$\begin{aligned} \pi_i &= P(S_1 = i) \\ a_{ij} &= P(S_{t+1} = j | S_t = i) \\ b_{ik} &= P(O_t = k | S_t = i) \end{aligned}$$

- E-step in HMMs must compute:

$$\begin{aligned} & P(S_1 = i | o_1, o_2, \ldots, o_T) \\ & P(S_{t+1} = j, S_t = i | o_1, o_2, \ldots, o_T) \\ & P(O_t = k, S_t = i | o_1, o_2, \ldots, o_T) = I(o_t, k)\, P(S_t = i | o_1, o_2, \ldots, o_T) \end{aligned}$$

$\underbrace{\text{special case of below } (t=1)}$

Forward-backward algorithm for inference in HMMs

- Summary of E-step:

$$
\begin{aligned}
P(S_t = i | o_1, \ldots, o_T) &= \frac{\alpha_{it}\, \beta_{it}}{\sum_j \alpha_{jt}\, \beta_{jt}} \\
P(S_t = i, S_{t+1} = j | o_1, \ldots, o_T) &= \frac{\alpha_{it}\, a_{ij}\, b_j(o_{t+1})\, \beta_{j,t+1}}{\sum_k \alpha_{kt}\, \beta_{kt}}
\end{aligned}
$$

EM algorithm for HMMs

- CPTs to re-estimate:

$$\begin{aligned} \pi_i &= P(S_1 = i) \\ a_{ij} &= P(S_{t+1} = j | S_t = i) \\ b_{ik} &= P(O_t = k | S_t = i) \end{aligned}$$

- M-step updates:

$$\begin{aligned} \pi_i &\leftarrow P(S_1 = i | o_1, o_2, \ldots, o_T) \\ a_{ij} &\leftarrow \frac{\sum_t P(S_{t+1} = j, S_t = i | o_1, o_2, \ldots, o_T)}{\sum_t P(S_t = i | o_1, o_2, \ldots, o_T)} \\ b_{ik} &\leftarrow \frac{\sum_t I(o_t, k) P(S_t = i | o_1, o_2, \ldots, o_T)}{\sum_t P(S_t = i | o_1, o_2, \ldots, o_T)} \end{aligned}$$

(for one sequence of observations)

Time complexity of HMM computations

T	length of observation sequence $(o_1, o_2, \ldots, o_T)$
n	cardinality of state space $s_t \in \{1, 2, \ldots, n\}$
m	cardinality of observation space $o_t \in \{1, 2, \ldots, m\}$

- **All of the following computations are $O(n^2 T)$:**

  (a) computing the likelihood $P(o_1, o_2, \ldots, o_T)$

  (b) decoding $\textbf{argmax}_{s_1,\ldots,s_T} P(s_1, \ldots, s_T | o_1, \ldots, o_T)$

  (c) re-estimating $\{\pi_i, a_{ij}, b_{ik}\}$ by one update of EM

  (d) updating beliefs $P(S_t = i | o_1, \ldots, o_t)$ for T steps

Reinforcement Learning

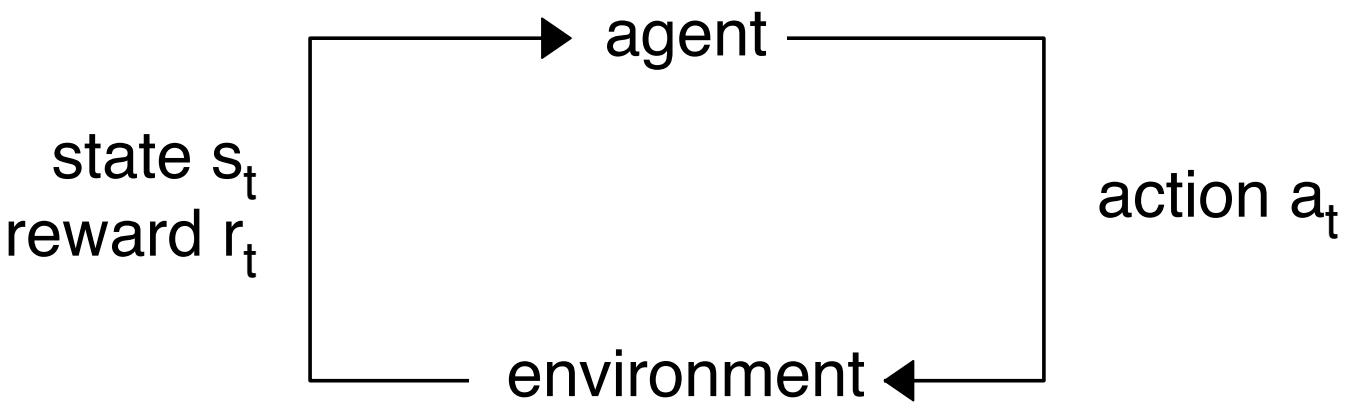
Reinforcement learning (RL)

How can autonomous decision-making agents learn from experience in the world?

Challenges of RL

1. How to learn in noisy, uncertain environments?
2. How to learn from evaluative (versus instructive) feedback?
3. When to explore, versus when to exploit?
4. How to learn from delayed (versus immediate) rewards?
5. How to navigate complex worlds with tractable models?
6. How to prove computational guarantees (e.g., convergence, optimality, efficiency)?

A probabilistic framework for RL



How do we formalize this process?

How do we handle uncertainty?

We define a Markov decision process.

Definition

A Markov decision process (MDP) is defined by the following:

- A **state space** $\mathcal{S}$ with states $s \in \mathcal{S}$
- An **action space** $\mathcal{A}$ with actions $a \in \mathcal{A}$
- **Transition probabilities**

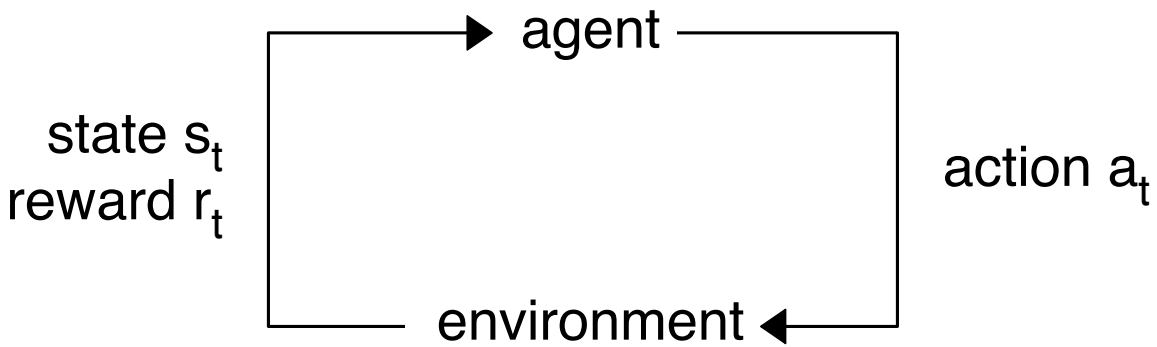
  $$P(s'|s,a) \;=\; P(S_{t+1}=s'|S_t=s, A_t=a)$$

  that indicate, at any time t, how frequently an agent moves from state s to state s' after taking action a
- A **reward function** $R(s,s',a)$, providing immediate feedback when the agent takes action a in state s and moves to state s'.

  Rewards are **scalar**: the higher, the better.

$$\boxed{\text{MDP} \;=\; \{\mathcal{S}, \mathcal{A}, P(s'|s,a), R(s,s',a)\}}$$

Markov assumptions



1. Conditional independence

$$P(S_{t+1}=s'|S_t=s,A_t=a)$$

$$= \quad P(S_{t+1}=s'|S_t=s,A_t=a,S_{t-1},A_{t-1},S_{t-2},A_{t-2},\ldots)$$

2. Transition probabilities are constant over time:

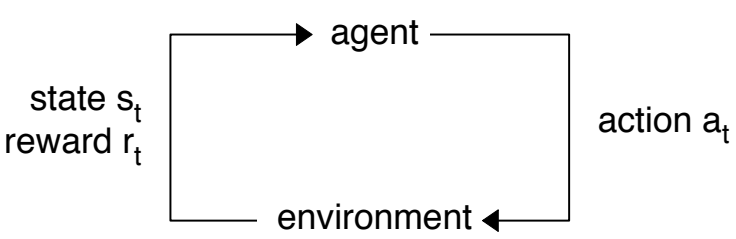
$$P(S_{t+1}=s'|S_t=s,A_t=a) \;=\; P(\underbrace{S_{t+1+\tau}=s'|S_{t+\tau}=s,A_{t+\tau}=a}_{\text{shifted by } \tau})$$

Simplifications for CSE 150A/250A

1. State space is discrete and finite: $\mathcal{S} = \{1, 2, \ldots, |\mathcal{S}|\}$.
2. Action space is discrete and finite: $\mathcal{A} = \{1, 2, \ldots, |\mathcal{A}|\}$.
3. Rewards depend only on the state: $R(s, s', a) = R(s)$.
4. Rewards are bounded: $\max_s |R(s)| < \infty$.
5. Rewards are deterministic.

$$\boxed{\text{MDP} = \{\mathcal{S}, \mathcal{A}, P(s'|s, a), R(s)\}}$$

Example: board games (with dice)



$s \in \mathcal{S}$ board position and results of roll of dice

$a \in \mathcal{A}$ one of any allowed moves

$$R(s) = \begin{cases} +1 & \text{if agent wins the game} \\ -1 & \text{if agent loses the game} \\ 0 & \text{for all preceding board positions} \end{cases}$$

$$P(s'|s,a) \sim \begin{cases} \text{agent moves} \\ \text{opponent rolls dice} \\ \text{opponent moves} \\ \text{agent rolls dice} \end{cases}$$

Activity: Build & Train a Simple RL Agent (Nim)

The Game (Nim):

- Start with 6 objects
- On your turn, you can take 1 or 2 objects.
- Last person to take an object loses.

Training

During the game:

- Agent plays first. When it is the agent's turn, draw a slip from the correct state.
- Leave the drawn slip next to that state.

After the game ends:

- **If the agent won:** fold and put all drawn slips back into their states.
- **If the agent lost:** discard the slip for the agent's last action. If a state would become empty, keep its last slip and instead discard the slip from the previous agent action. Return all earlier slips to their states.

Decision-making in MDPs

- **Definition**

  A **policy** $\pi : \mathcal{S} \to \mathcal{A}$ is a mapping of states to actions.
  In this class we will only consider deterministic policies.

- **Number of policies**

  If there are $|\mathcal{A}|$ possible actions in each of $|\mathcal{S}|$ states, then there are *combinatorially* many policies:

$$\#\text{ policies } = |\mathcal{A}|^{|\mathcal{S}|}$$

- **Experience under policy π**

$$\begin{array}{rccccccc} \text{state} & s_0 & \xrightarrow{\pi(s_0)} & s_1 & \xrightarrow{\pi(s_1)} & s_2 & \cdots \\ \text{reward} & r_0 & & r_1 & & r_2 & \cdots \end{array}$$

Transitions occur with probabilities $P(s'|s, \pi(s))$.

How to measure long-term return?

1. **Finite-horizon return**

$$\text{return} = \frac{1}{T}(r_0 + r_1 + \cdots + r_{T-1}) \quad \text{for a } T\text{-step horizon}$$

2. **Undiscounted return with infinite horizon**

$$\text{return} = \lim_{T \to \infty} \left[\frac{1}{T} \sum_{t=0}^{T-1} r_t \right]$$

These are the most obvious ways to accumulate rewards.
But they are not the most commonly used in practice ...

How to measure long-term return? (con't)

3. **Discounted return with infinite horizon**

   Let $\gamma \in [0, 1)$ denote the so-called **discount factor**.
   Then define

$$\text{return} = r_0 + \gamma r_1 + \gamma^2 r_2 + \gamma^3 r_3 + \cdots = \sum_{t=0}^{\infty} \gamma^t r_t$$

   When $\gamma \ll 1$, future rewards are heavily discounted.
   These returns can be optimized by short-sighted agents.

   When γ is close to 1, future rewards are lightly discounted.
   These returns can only be optimized by far-sighted agents.

Motivation for $\gamma \in [0, 1)$

Psychologist: *Why discount rewards from the distant future?*
Economist: *Why favor investments with short-term payoffs?*

1. **Intuition**

   Many models are only approximations to the real world; we should not attempt to extrapolate them indefinitely.

2. **Mathematical convenience**

   Discounted returns lead to simple iterative algorithms with strong guarantees of convergence.

What to optimize?

The discounted return $\sum_{t=0}^{\infty} \gamma^t r_t$ is a random variable.
But we can try to optimize its expected value:

$$\mathrm{E}^{\pi}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t) \,\middle|\, s_0 = s\right]$$

the expected value of the discounted infinite-horizon return, starting in state s at time $t=0$, and following policy π.

Maximizing the expected return is:

- generally wiser than maximizing the best-case return,
- but not as robust as minimizing the worst-case return.

Test your understanding

- Learning from experience in the world

Which of the following is **NOT** part of the definition of an MDP?

A. A set of states $\mathcal{S}$ with $s \in \mathcal{S}$

B. Transition probabilities: $P(s'|s) = P(S_{t+1}=s'|S_t=s)$

C. An action space $\mathcal{A}$ with actions $a \in \mathcal{A}$

D. Reward function R(S)

E. None of the above.

Test your understanding

When calculating the long-term value of a state sequence over time, rewards for states in the near future count more than rewards for states in the distant future.

True (A) or False (B)?

That’s all folks!