Literature review: Reinforcement Learning

Daniel Hernandez

1 Environment models

Even though Markov Decision Processes are the most famous mathematical structure used to model an environment in reinforcement learning, there are other types of possible models for RL environment which act as extensions to vanilla MDPs. These mainly relax assumptions about the state space, the action space, the observability of the environment state, and the reward function. This section concerns itself to defining these extensions, and making links between them. This is not an exhaustive list of all possible mathematical models used to represent environments in the RL literature. However, these are some of the most used or fundamental models in the field, on which the majority of the research is conducted, and on top of which most niche extensions are built. Table 1 features all the environment models discussed in this section as well as their differences with respect with MDPs.

Model	Partial observability	Multi-agent	Multiple Reward functions	Delayed actions
SMDP	×	×	×	✓
POMDP	✓	×	×	×
MMDP	×	✓	×	×
dec-POMDP	✓	✓	×	×
Markov Game	×	✓	✓	×

Table 1: Properties of various environment models with respect to classical Markov Decission Processes.

1.1 Semi Markov Decision Process (SMDP)

As stated in (Barto, 2003), in an MDP, only the sequential nature of the decision process is relevant, not the amount of time that passes between decision points. Semi Markov Decision Processes do not assume thatthe time elapsed in between decision points, also known as decision stages, is constant. Every action taken in an SMDP has an assigned delay τ, known as *holding time*. When an action with holding time τ is taken state s_t, the agent waits for τ time before the action is executed and the next decision point s_{t+1} is reached. The agent then recieves a cumulative reward obtained throughout the elapsed timesteps, $r_t = \int_{t'=0}^{\tau} r_{t+t'}$(TODO: not sure about this...). The time until the next decision point τ can only depend on the action a and state t and thus τ is independent of the history of the environment. SMDPs can also be used for real-valued time systems instead of discretely timed environments. This holding time allows for a gap in time between sensorial input reaching the agent and the agent's action being executed on the environment.

This type of process is considered Semi Markovian because as the holding time is elapsing, the agent cannot know how the system is evolving. Thus, in order to determine when the next state (decision point) will be reached, it is necessary to know how much time has elapsed, introducing temporal dependency, breaking the Markov property. This is formally described as: the probability of reaching state s_{t+1} depends only on s_t and action a_t with associated holding time τ. Once the action a_t has been decided, estimating when the agent will recieve state $s_{t+\tau}$ depends on how much time has elapsed since the action a_t was decided.

A Semi Markov Decision Process is defined by a 5-element tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}(\cdot, \cdot|\cdot, \cdot), \mathcal{R}(\cdot, \cdot, \cdot, \cdot), \gamma)$:

- $\mathcal{S}, \mathcal{A}$ and γ express the same concepts as in classical MDPs.
- $\mathcal{P}(s', \tau|s, a)$, where $s', s \in \mathcal{S}, \tau \in \mathbb{N}, a \in \mathcal{A}$, is the transition probability function. Which states the probability of transitioning to state s‘ after a holding time of τ.

- $\mathcal{R}(s, a, s', \tau)$, where $s', s \in \mathcal{S}, a \in \mathcal{A}$, is the reward function. It represents the expected reward of deciding on action a on state s with an assigned holding time of τ timesteps and landing on state s'.

A useful properties of SMDPs is that they can be reduced to regular MDPs through the *data-transformation method* (Piunovskiy and Zhang, 2012). This introduces the possibility of using MDP solving methods to solve SMDPs. SMDPs have recieved a lot of attention in the field hierarchical learning, especially with regards to options (Sutton and Barto, 1998).

1.2 Partially Observable Markov Decision Process (POMDP)

In an MDP, the internal representation of the environment is the same representation that the agent receives at every timestep. POMDPs introduce the idea that what the agent observes at every timestep t is only a partial representation o_t of the real environment state s_t. This partial observation o_t alone is not enough to reconstruct the real environment state s_t, which entails that $o_t \subset s_t$. A Partially Observable Markov Decision process is defined by a 6-element tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}(\cdot|\cdot,\cdot), \mathcal{R}(\cdot,\cdot), \Omega, \mathcal{O}(\cdot|\cdot,\cdot), \gamma)$:

- $\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}$ and γ express the same concepts as in classical MDPs.
- Ω is the set of all possible agent observations. Notably $\Omega \subset \mathcal{S}$, meaning that some of the state properties may never be available to the agent.
- $\mathcal{O}(o|s, a)$, where $o \in \mathcal{O}, s \in \mathcal{S}, a \in \mathcal{A}$, represents the probability of the agent recieving observation o after executing action a in state s.

In an POMDP the goal of the environment is not to find an optimal policy π^* *conditioned* on the history of environment observations which will maximize the expected cumulative reward. The agent samples actions from its policy, which is no longer conditioned on the state of the environment, as the agent does not have access to it, but rather it is conditioned on the sequence of the observations that the agent has obtained so far, $a_t \sim \pi(o_{\leq t})$. This goal is formalized as:

$$\pi = \max_{\pi} \mathbb{E}_{s_0 \sim \rho_0, s \sim \xi, a \sim \pi(\cdot|o_{\leq t})}[\sum_{t=0}^{T} \gamma r_t] \tag{1}$$

A POMDP can be reduced to an MDP iff, for all timesteps t the agent's observation o_t and the environment state s_t are equal $o_t = s_t$.

1.3 Multi-agent Markov Decision Process (MMDP)

A major shortcoming of MDPs is that they assume stationary environments, which by definition entails that the environment does not change over time. This assumption makes MDPs unsuitable for modelling multi-agent environments. Agents must be considered as non-stationary parts of the environment, because the policies that define their behaviours change over time through the course of learning, breaking the environment stationarity assumption.

Boutilier (1996) introduces Multi-agent Markov Decision Processes (MMDPs) as framework to study coordination mechanisms (TODO: give as an example what usage they gave). MMDPs feature multiple agent policies, each of them submitting an individual action every timestep, which is executed as a joint action by the environment, producing a new state via the transition probability function.

A Multi-agent Markov Decission Process featuring k agents is defined by a 5-element tuple (TODO fix dumb latex indentation error) $(\mathcal{S}, \mathcal{A}_{1..k}, \mathcal{P}(\cdot,\cdot|\cdot,\cdot), \mathcal{R}(\cdot,\cdot), \gamma)$:

- $\mathcal{S}$ and γ express the same concepts as in classical MDPs.
- $\mathcal{A}_{1..k}$ is a collection of action sets, one for each agent in the environment, with $\mathcal{A}_i$ corresponding to the action set of the ith agent.
- $\mathcal{P}(s'|s, \mathbf{a})$, where $s \in \mathcal{S}$, $\mathbf{a} = \{a_1, \ldots, a_k\}$ and $a_1 \in \mathcal{A}_1, \ldots, a_k \in \mathcal{A}_k$, represents the probability transition function. It states the probability of transitioning from state s to state s' after executing the *joint* action $\mathbf{a}$. The joint action is a vector containing the action performed by every agent at a certain timestep.

- $\mathcal{A}_{1..k}$ is a collection of action sets, one for each agent in the environment, with $\mathcal{A}_i$ corresponding to the action set of the ith agent.
- $\mathcal{R}(s, \mathbf{a})$, where $s \in \mathcal{S}$, $\mathbf{a} = \{a_1, \dots, a_k\}$ and $a_1 \in \mathcal{A}_1, \dots, a_k \in \mathcal{A}_k$, represents the reward function.

MMDPs can be thought as k-person stochastic games in which the payoff function is the same for all agents.

1.4 Decentralized Markov Decision Process (dec-POMDP)

Dec-POMDPs form a framework for multiagent planning under uncertainty (Oliehoek and Amato, 2014). This uncertainy comes from two sources. The first one being the partial observability of the environment, the second one stemming from the uncertainy that each agent has over the other agent's policies. It is the natural multi-agent generalization of POMDPs, introducing multi-agent concepts analogous to that of MMDPs. They are considereded *decentralized* because there is no explicit communication between agents. Agents do not have the explicit ability of sharing their observations and action choices with each other. Every agent bases its decision purely on its own individual observations. On every timestep each agent chooses an action simultaneously and they are all collectively submitted to the environment. As in MMDPs, all agents share the same reward function, making the nature of dec-POMDPs collaborative. A decentralized Partially Observable Markov Decision Process is defined by an 8-element tuple $(I, \mathcal{S}, \mathcal{A}_{1..k}, \mathcal{P}(\cdot|\cdot,\cdot), \mathcal{R}(\cdot,\cdot), \Omega_{1..k}, \mathcal{O}(\cdot|\cdot,\cdot), H)$:

- $\mathcal{S}$ expresses the same concepts as in classical MDPs.
- $I = \pi_i, ..\pi_k$ is the set of all agent policies.
- $\mathcal{A}_{1..k}$ is a collection of action sets, one for each agent in the environment, with $\mathcal{A}_i$ corresponding to the action set of the ith agent.
- $\mathcal{P}(s'|s, \mathbf{a})$, where $s \in \mathcal{S}$, $\mathbf{a} = \{a_1, \dots, a_k\}$ and $a_1 \in \mathcal{A}_1, \dots, a_k \in \mathcal{A}_k$, represents the probability transition function. It states the probability of transitioning from state s to state s' after executing the *joint* action $\mathbf{a}$. The joint action is a vector containing the action performed by every agent at a certain timestep.
- $\mathcal{R}(s, \mathbf{a})$, where $s \in \mathcal{S}$, $\mathbf{a} = \{a_1, \dots, a_k\}$ and $a_1 \in \mathcal{A}_1, \dots, a_k \in \mathcal{A}_k$, represents the reward function.
- $\Omega_{1..k}$ represents the joint set of all agent observations, with Ω_i representing the set of all possible observations for the ith agent.
- $\mathcal{O}(\mathbf{o}|s, \mathbf{a})$, where $\mathbf{o} = \{o_1, \dots, o_k\}$, $o_1 \in \Omega_1, \dots, o_k \in \Omega_k$, $s \in \mathcal{S}$, $\mathbf{a} = \{a_1, \dots, a_k\}$ and $a_1 \in \mathcal{A}_1, \dots, a_k \in \mathcal{A}_k$, represents the probability of observing the joint observation vector $\mathbf{o}$, containing an observation for each agent, after executing the joint action $\mathbf{a}$ in state s.
- $H \in \mathbb{N}$ represents the finite time horizon, the number of steps over which the agents will try to maximize their cummulative reward. It serves a similar purporse to the more common discount factor γ in that they are both used as a variance-bias trade off and are needed for proofs of convergence over infinitely long running tasks.

A dec-POMDP featuring a single agent, $|I| = 1$, can be treated as a POMDP. When theenvironment features full observability, the term dec-MDP is used.

1.5 Markov Game

Owen and Owen (1982) first introduced the notion of a Markov Game. Markov Games also serve to model multi-agent environments. They came to be as a crossbreed between game theoretic structures such as extended-form games and Markov Decision Processes. A Markov Game with k different agents is denoted by a 5-element tuple $(\mathcal{S}, \mathcal{A}_{1..k}, \mathcal{P}(\cdot|\cdot,\cdot), \mathcal{R}_{1..k}(\cdot,\cdot), \gamma)$

- $\mathcal{S}$ and γ express the same concepts as classical MDPs.
- $\mathcal{A}_{1..k}$ is a collection of action sets, on for each agent in the environment, with $\mathcal{A}_i$ corresponding to the action set of the ith agent.

- $\mathcal{P}(s'|s, \mathbf{a})$, where $s \in \mathcal{S}$, $\mathbf{a} = \{a_1, \ldots, a_k\}$ and $a_1 \in \mathcal{A}_1, \ldots, a_k \in \mathcal{A}_k$, represents probability transition function. It states the probability of transitioning from state s to state s' after executing the *joint* action $\mathbf{a}$. The joint action represents all of the actions that were executed at timestep t.
- $\mathcal{R}(s_t, \mathbf{a_t}) \in \mathbb{N}^k$, where $s_t \in \mathcal{S}$, $\mathbf{a_t} = \{a_1, \ldots, a_k\}$ and $a_1 \in \mathcal{A}_1, \ldots, a_k \in \mathcal{A}_k$, represents the reward function. $\boldsymbol{r_t} \in \mathbb{N}^k$ is the reward vector. The reward $r_i \in \boldsymbol{r_t}$ is the reward that the ith agent will obtain after the joint action vector $\mathbf{a}$ is executed in state s.

Each agent independently tries to maximize its expected discounted cumulative reward, $\mathbb{E}[\sum_{j=0}^{\infty} \gamma^j r_{i,t+j}]$, where $r_{i,t+j}$ is the reward obtained by agent i at time $t + j$. (TODO: introduce maximization policy objective in a similar fashion as done with POMDPs)

Markov Games have several important properties Owen and Owen (1982); Littman (1994). Like MDP's, Every Markov game features an optimal policy for each agent. Unlike MDPs, these policies may be *stochastic*. An intuitive advantage of stochastic policies stems from the agent's uncertainty about the opponent's pending moves. On top of this, sthocastic policies make it difficult for opponents to "second guess" the agent's action, which makes the policy less exploitable.

When the number of agents in a Markov Game is exactly 1, the Markov Game can be considered an MDP. When $|\mathcal{S}| = 1$, the environment can be considered a normal-form game from game theory literature. Doning a game theory hat, γ can be thought of as the probability of the game finishing next round. If all agents shared the same reward function, the Markov Game is reduced to an MMDP.

References

Barto, A. G. (2003). Recent Advances in Hierarchical Reinforcement Learning Markov and Semi-Markov Decision Processes. *Most*, 13(5):1–28.

Boutilier, C. (1996). Planning, learning and coordination in multiagent decision processes. *Proceedings of the 6th conference on Theoretical aspects of rationality and knowledge*, pages 195–210.

Littman, M. L. (1994). Markov games as a framework for multi-agent reinforcement learning. *Machine Learning Proceedings 1994*, pages 157–163.

Oliehoek, F. A. and Amato, C. (2014). Best Response Bayesian Reinforcement Learning for Multiagent Systems with State Uncertainty. *AAMAS Workshop on Multiagent Sequential Decision Making Under Uncertainty, MSDM 2014*, (May).

Owen, G. and Owen, G. (1982). Game Theory. *Collection.*

Piunovskiy, A. and Zhang, Y. (2012). The Transformation Method for Continuous-Time Markov Decision Processes. *Journal of Optimization Theory and Applications*, 154(2):691–712.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning: An Introduction.*