
Sequential Decision-Making Under Uncertainty: Introduction to Reinforcement Learning

Siddharth Srivastava, Ph.D.
Assistant Professor
Arizona State University

Markov Decision Processes (MDPs)

S set of states

– E.g., $At(1,1)$

A set of actions

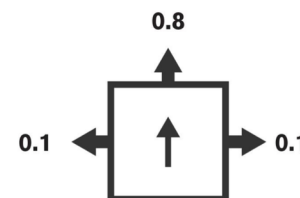
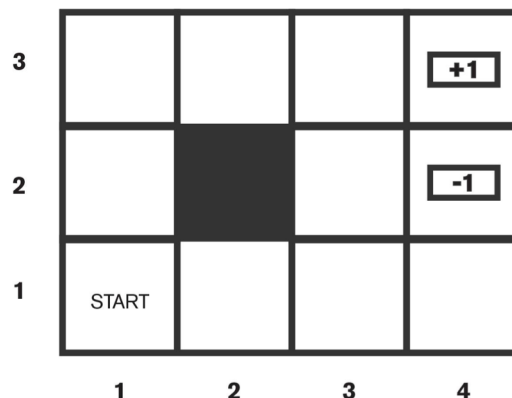
T transition model

– $P(s'|s, a) = T(s, a, s')$ }

$R: S \rightarrow \mathbb{R}$ reward, or utility function

Agent can “drift”, end up in unintended states

Solutions take the form of policies: $\pi: S \rightarrow A$



What if the Agent Encounters a New Environment?

S set of states

– E.g., $At(1,1)$

A set of actions

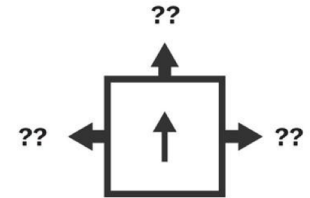
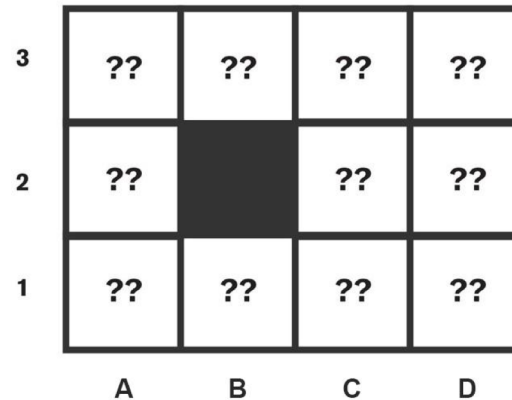
T transition model

– $P(s'|s, a) = T(s, a, s')$

$R: S \rightarrow \mathbb{R}$

Unknown

Map not known to the agent
(or partially known)



Adapting to new situations is an essential component of intelligence

How would we want an AI agent deal with it?

Ideas for Dealing with Change

| Transfer + adapt known knowledge, policy

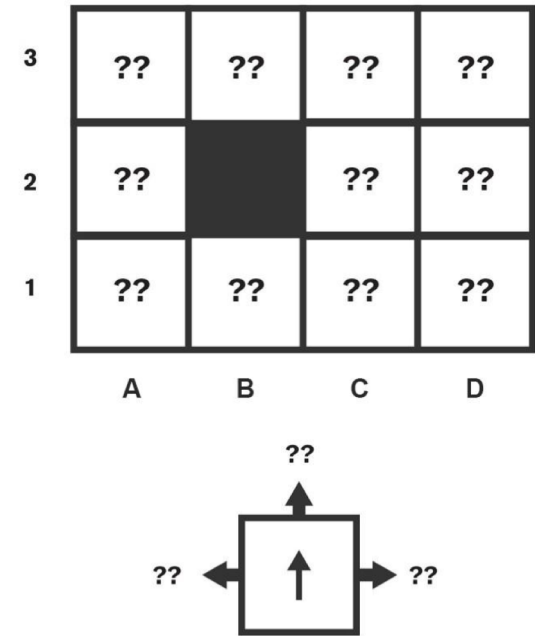
- Try the existing policy
- Recompute policy with any changes noticed

| Transfer learned principles

- Walls are impassable

| Learn new environment model, then plan

| Learn while acting in the environment



Map not known to the agent
(or partially known)

Integrating Planning and Learning



| **Reinforcement learning**: learning to act based on online feedback from the environment

| **Fundamental trade-off:**

- **Explore**: experiment, learn about the environment
- **Exploit**: settle on the learned knowledge, use current best policy to cut (possible) losses

| **Particularly useful when agent acts in the *real* environments (pure RL)**

- Real as in not in a simulator (model free)
- Practically limited to controlled settings

| **Use models (e.g. simulators/analytical descriptions) when possible to reduce risk**

Progression Towards RL



| Next few slides:

- Ideas for planning in unknown environments, leading up to RL

Idea 1: Learn Model First, Then Plan



- | First, learn T , R based on available data

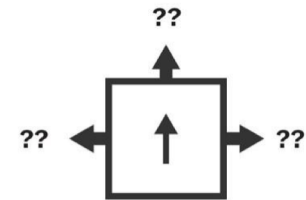
- | Next, compute an optimal policy

Example of Model Learning

| Training data compiled using a given policy (transitions from A1):

- A1, →, B1
- A1, →, A2
- A1, →, B2
- A1, →, A2
- A1, →, B1
- A1, →, B1
- A1, →, A2

3	??	??	??	??
2	??		??	??
1	??	??	??	??
	A	B	C	D



| Learned model:

$$P(B1|A1, \rightarrow) = \frac{3}{7}, P(B2|A1, \rightarrow) = \frac{1}{7},$$
$$P(A2|A1, \rightarrow) = \frac{3}{7}$$

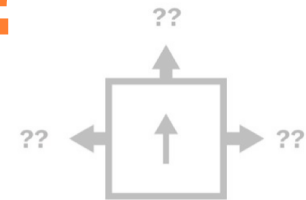
Example of Model Learning

| Training data compiled using a given policy (transitions from A1):

- A1, \rightarrow , B1
- A1, \rightarrow , A2
- A1, \rightarrow , B2
- A1, \rightarrow , A2
- A1, \rightarrow , B1
- A1, \rightarrow , B1
- A1, \rightarrow , A2

3	??	??	??	??
2	??		??	??
1	??	??	??	??
	A	B	C	D

Assumptions?



| Learned model:

$$P(B1|A1, \rightarrow) = \frac{3}{7}, P(B2|A1, \rightarrow) = \frac{1}{7},$$
$$P(A2|A1, \rightarrow) = \frac{3}{7}$$

Example of Model Learning

Assumptions?

- Set of states is known (representable in some form)
- Set of possible actions is known
- Reward function is deterministic
- Transition probabilities are stationary
- Environment is Markovian

We are still in the MDP framework...

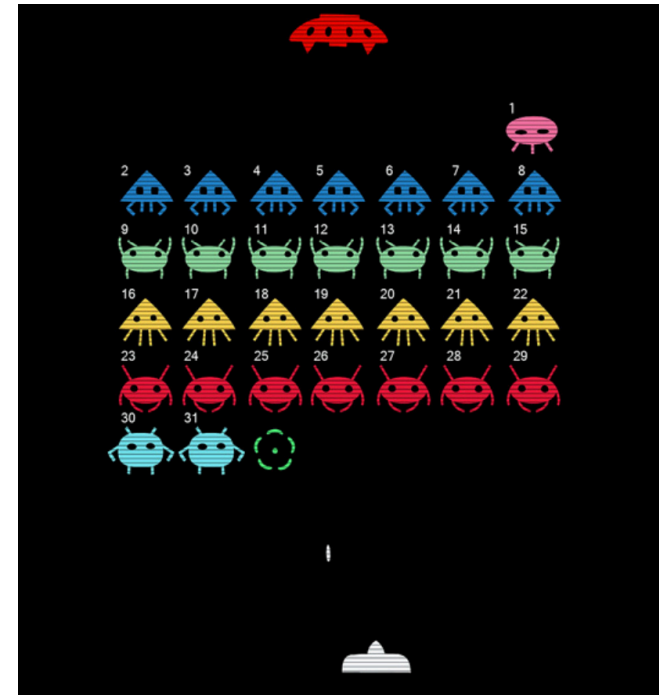
All approaches we consider for SDM (and most current approaches) make these assumptions

Analysis of Learning Followed by Planning

| Learn T, R based on available data

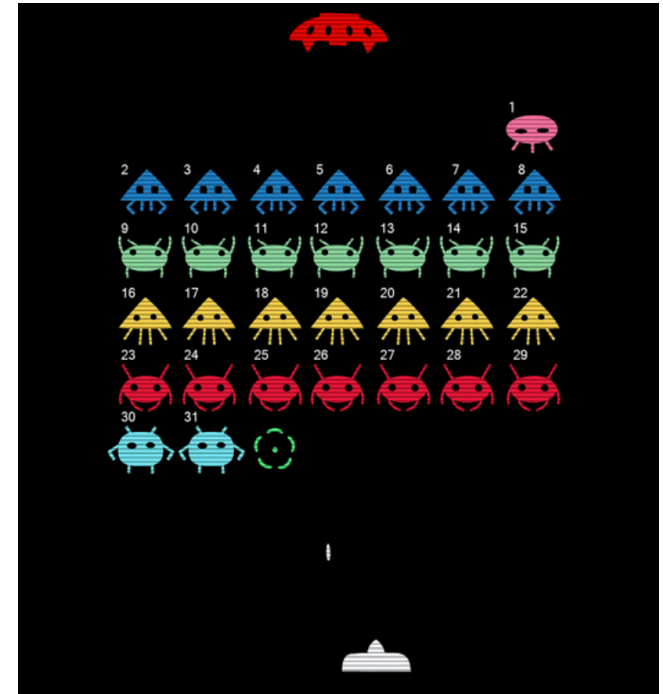
| Advantages:

- Easy to transfer to new situations if some information about changes is available
- E.g., household robotics:
 - Robot learned to deliver coffee when doors were open
 - New environment: doors closed
 - Can focus learning to opening doors
- Video games
 - New reward function: need to keep last row alive for 10s!



Limitations of Model Learning

- | Agent needs time to learn
- | During learning, agent pauses its goal-achievement activity
- | A bit like training for a new sport: no tournaments before reaching a level of competence



Idea 2:

Monte Carlo Policy Evaluation (“Passive” RL)

| Can we learn a policy directly?

| Suppose the AI agent is a self-driving car

- Being driven by a human driver
- Essentially being given a policy

| Still doesn't know **T, R**

| **Goal:** Learn the value function V^π directly, without learning T or R



Why Can't we Use Policy Evaluation?



$$V_0^\pi(s) = 0$$

$$V_{k+1}^\pi(s) = \sum_{s'} T(s, \pi(s), s') [R(s) + \gamma V_k^\pi(s')]$$

Monte Carlo Policy Evaluation

| Break up training into episodes

- Recall: reward collected before moving to next state

| Episode 1:

- $A1, \rightarrow, B1, 0$
- $B1, \leftarrow, B1, 0$
- $B1, \leftarrow, A1, 0$
- $A1, \uparrow, A2, -10$
- $A2, \uparrow, A3, 20$
- $A3, \rightarrow, A4, -5$
- $A4, \rightarrow, A4, 0$
- $A1, \leftarrow A4, 0$

Initialize:

$\pi \leftarrow$ policy to be evaluated

$V \leftarrow$ an arbitrary state-value function

$Returns(s) \leftarrow$ an empty list, for all $s \in \mathcal{S}$

Repeat forever:

(a) Generate an episode using π

(b) For each state s appearing in the episode:

$R \leftarrow$ return following the first occurrence of s

Append R to $Returns(s)$


$V(s) \leftarrow \text{average}(Returns(s))$

Sutton & Barto, RL, 1st ed.

Monte Carlo Policy Evaluation

| Break up training into episodes

| Episode 1:

- $A1, \rightarrow, B1, 0$
 - $B1, \leftarrow, B1, 0$
 - $B1, \leftarrow, A1, 0$
 - $A1, \uparrow, A2, -10$
 - $A2, \uparrow, A3, 20$
 - $A3, \rightarrow, A4, -5$
 - $A4, \rightarrow, A4, 0$
 - $A1, \leftarrow A4, 0$
- 

Initialize:

$\pi \leftarrow$ policy to be evaluated

$V \leftarrow$ an arbitrary state-value function

$Returns(s) \leftarrow$ an empty list, for all $s \in \mathcal{S}$

Repeat forever:

(a) Generate an episode using π

(b) For each state s appearing in the episode:

$R \leftarrow$ return following the first occurrence of s

Append R to $Returns(s)$

$V(s) \leftarrow \text{average}(Returns(s))$

[Sutton & Barto, RL, 1st ed.]

$$V(A1) = 5; V(A2) = 15; V(A3) = -5$$

Analysis of Monte Carlo Policy Evaluation

| **Advantage:** can focus on a subset of states; learn while collecting data

| **Limitation:** doesn't utilize state-reachability

- Suppose in the next episode we get one sample:
- $C1, \leftarrow, B1, 0$
- MC evaluation: $V(C1) = 0$

| **We already have training data from $B1$**

- $V(C1)$ could have used previous recorded returns from $B1...$

3	??	??	??	??
2	??		??	??
1	??	??	??	??
	A	B	C	D