ECEN 743: Reinforcement Learning

Markov Decision Process (MDP): Introduction

Dileep Kalathil
Assistant Professor
Department of Electrical and Computer Engineering
Texas A&M University

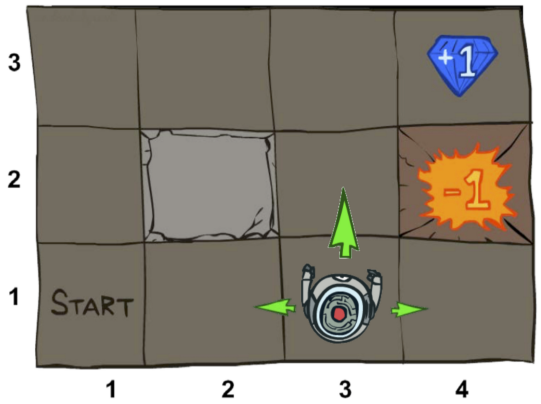
References

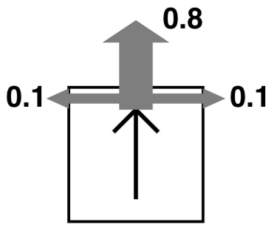
- [SB, Chapter 3-4]
- [DB, Chapter 2]
- [AJKS, Chapter 1]
- Puterman, *Markov Decision Process*, Chapter 1-3, 5-6

- Acknowledgment: Some figures for this lecture are taken from UC Berkeley CS188 course, with permission.

Grid World Example

- Agent lives in a grid world
- This world is stochastic

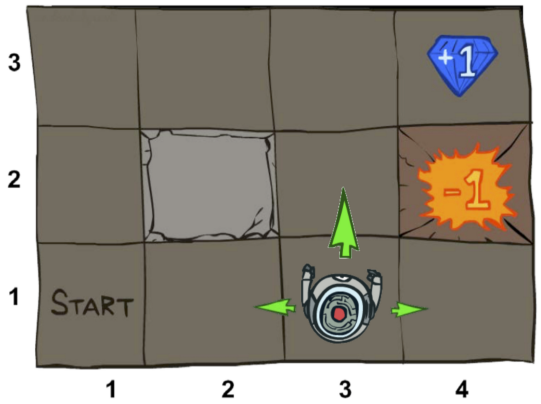


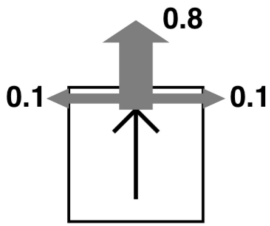


[1] **Figures are from** https://inst.eecs.berkeley.edu/~cs188/fa22/

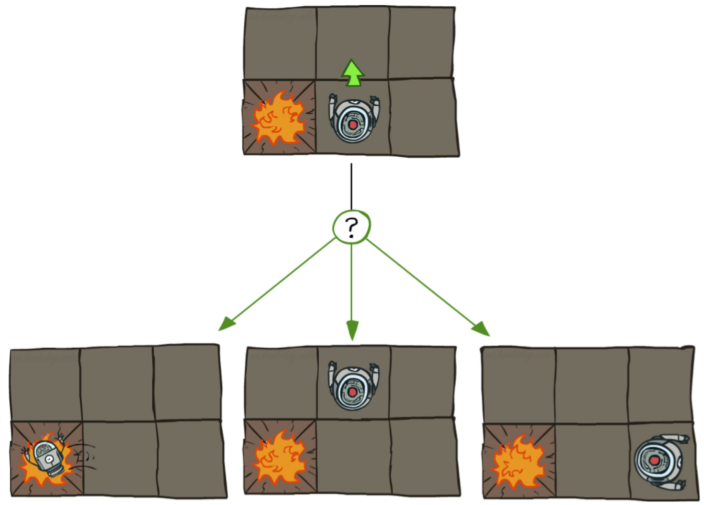
Grid World Example

- Agent lives in a grid world
- This world is stochastic





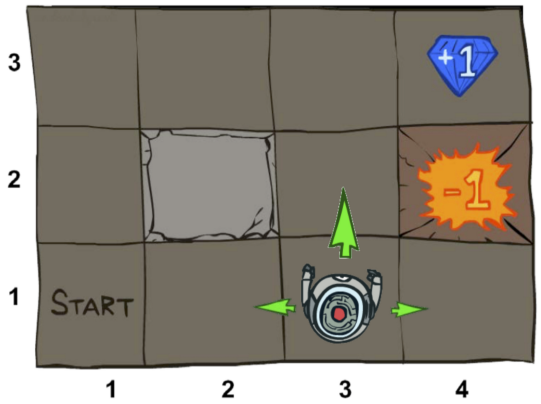
- Deterministic vs stochastic world:

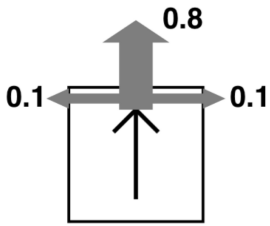


[1] **Figures are from** https://inst.eecs.berkeley.edu/~cs188/fa22/

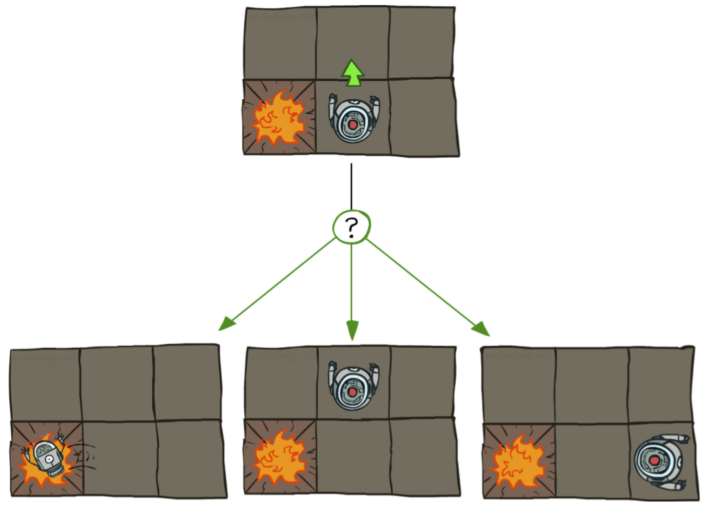
Grid World Example

- Agent lives in a grid world
- This world is stochastic





- Deterministic vs stochastic world:

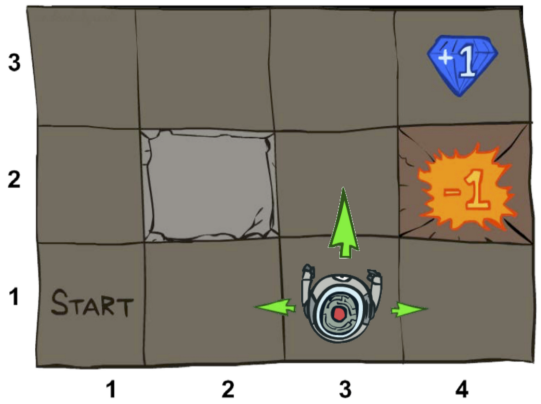


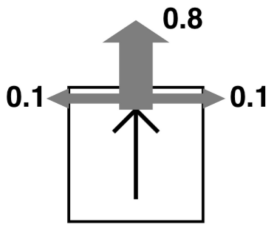
- Why do we need to model a stochastic world?

[1] **Figures are from** https://inst.eecs.berkeley.edu/~cs188/fa22/

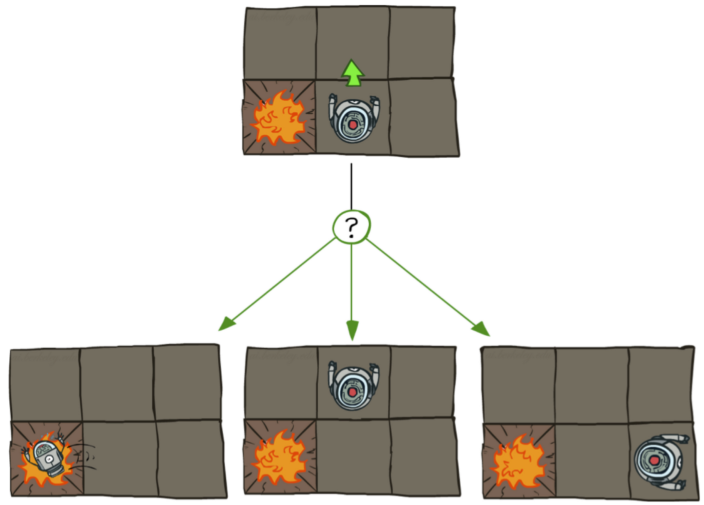
Grid World Example

- Agent lives in a grid world
- This world is stochastic





- Deterministic vs stochastic world:



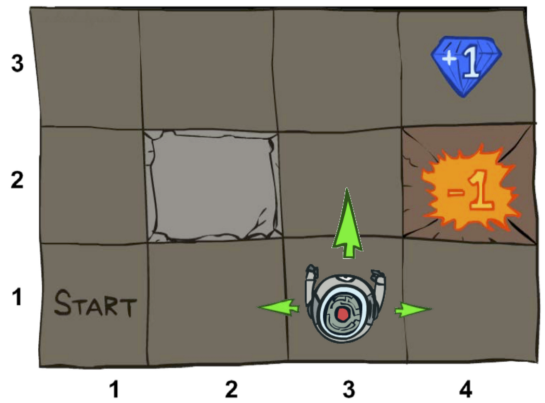
- Why do we need to model a stochastic world?
  - Inherent uncertainties in the system
  - Incomplete information about the system

.[1]

[1] **Figures are from** https://inst.eecs.berkeley.edu/~cs188/fa22/

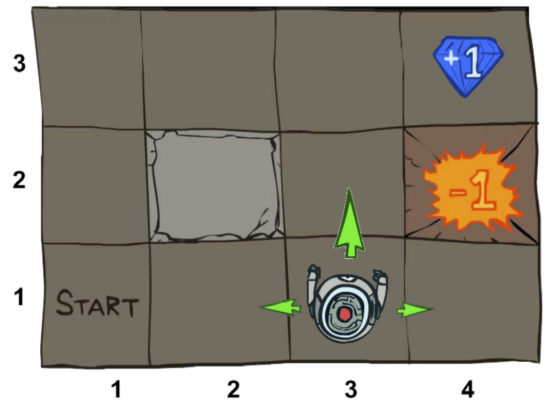
Markov Decision Process (MDP)

- A set of states $\mathcal{S}$ (state space)
- A set of action $\mathcal{A}$ (action space)
- Probability transition function P
  - $P(s'|s,a)$ represents the probability of moving from state s to state s' if action a is taken
  - Also called model of the system
- Reward function $r(s,a)$



Markov Decision Process (MDP)

- A set of states $\mathcal{S}$ (state space)
- A set of action $\mathcal{A}$ (action space)
- Probability transition function P
  - $P(s'|s,a)$ represents the probability of moving from state s to state s' if action a is taken
  - Also called model of the system
- Reward function $r(s,a)$



- MDP is an extremely useful mathematical formalism for modeling and solving many real-world control problems

Markov Property

- Markov property: this roughly means that given the present, the future and the past are independent

Markov Property

- Markov property: this roughly means that given the present, the future and the past are independent

$$\mathbb{P}(s_{t+1} = s' | s_t = \bar{s}_t, a_t = \bar{a}_t, \ldots, s_0 = \bar{s}_0, a_0 = \bar{a}_0) = \mathbb{P}(s_{t+1} = s' | s_t = \bar{s}_t, a_t = \bar{a}_t)$$

Markov Property

- Markov property: this roughly means that given the present, the future and the past are independent

$$\mathbb{P}(s_{t+1} = s' | s_t = \bar{s}_t, a_t = \bar{a}_t, \ldots, s_0 = \bar{s}_0, a_0 = \bar{a}_0) = \mathbb{P}(s_{t+1} = s' | s_t = \bar{s}_t, a_t = \bar{a}_t)$$

- In other words, if we know the current state and current action, history is irrelevant for predicting the future states

Markov Property

- Markov property: this roughly means that given the present, the future and the past are independent

$$\mathbb{P}(s_{t+1} = s' | s_t = \bar{s}_t, a_t = \bar{a}_t, \ldots, s_0 = \bar{s}_0, a_0 = \bar{a}_0) = \mathbb{P}(s_{t+1} = s' | s_t = \bar{s}_t, a_t = \bar{a}_t)$$

- In other words, if we know the current state and current action, history is irrelevant for predicting the future states
- Is the world Markov?

Markov Property

- Markov property: this roughly means that given the present, the future and the past are independent

$$\mathbb{P}(s_{t+1} = s' | s_t = \bar{s}_t, a_t = \bar{a}_t, \ldots, s_0 = \bar{s}_0, a_0 = \bar{a}_0) = \mathbb{P}(s_{t+1} = s' | s_t = \bar{s}_t, a_t = \bar{a}_t)$$

- In other words, if we know the current state and current action, history is irrelevant for predicting the future states
- Is the world Markov?
  - A very useful modeling assumption for a large class of real world problems!

Markov Property

- Markov property: this roughly means that given the present, the future and the past are independent

$$\mathbb{P}(s_{t+1} = s' | s_t = \bar{s}_t, a_t = \bar{a}_t, \ldots, s_0 = \bar{s}_0, a_0 = \bar{a}_0) = \mathbb{P}(s_{t+1} = s' | s_t = \bar{s}_t, a_t = \bar{a}_t)$$

- In other words, if we know the current state and current action, history is irrelevant for predicting the future states
- Is the world Markov?
  - A very useful modeling assumption for a large class of real world problems!
- Why is it useful?
  - No need to store the history (past states and actions) for the purpose of taking the future actions

Markov Property

- Markov property: this roughly means that given the present, the future and the past are independent

$$\mathbb{P}(s_{t+1} = s' | s_t = \bar{s}_t, a_t = \bar{a}_t, \ldots, s_0 = \bar{s}_0, a_0 = \bar{a}_0) = \mathbb{P}(s_{t+1} = s' | s_t = \bar{s}_t, a_t = \bar{a}_t)$$

- In other words, if we know the current state and current action, history is irrelevant for predicting the future states
- Is the world Markov?
  - A very useful modeling assumption for a large class of real world problems!
- Why is it useful?
  - No need to store the history (past states and actions) for the purpose of taking the future actions
  - Tremendous reduction in memory/computation

MDP Examples

MDP Modelling: Recycling Robot

- A mobile robot has the job of collecting empty soda cans in an office

MDP Modelling: Recycling Robot

- A mobile robot has the job of collecting empty soda cans in an office
- It has sensors for detecting cans, and an arm and gripper that can pick them up and place them in an onboard bin

MDP Modelling: Recycling Robot

- A mobile robot has the job of collecting empty soda cans in an office
- It has sensors for detecting cans, and an arm and gripper that can pick them up and place them in an onboard bin
- It runs on a rechargeable battery

MDP Modelling: Recycling Robot

- A mobile robot has the job of collecting empty soda cans in an office
- It has sensors for detecting cans, and an arm and gripper that can pick them up and place them in an onboard bin
- It runs on a rechargeable battery
- Objective: How to search for cans without depleting the battery?

MDP Modelling: Recycling Robot

- A mobile robot has the job of collecting empty soda cans in an office
- It has sensors for detecting cans, and an arm and gripper that can pick them up and place them in an onboard bin
- It runs on a rechargeable battery
- Objective: How to search for cans without depleting the battery?
- How do we model this as an MDP?

MDP Modelling: Recycling Robot

- A mobile robot has the job of collecting empty soda cans in an office
- It has sensors for detecting cans, and an arm and gripper that can pick them up and place them in an onboard bin
- It runs on a rechargeable battery
- Objective: How to search for cans without depleting the battery?
- How do we model this as an MDP?
  - What are the states and actions of the system?

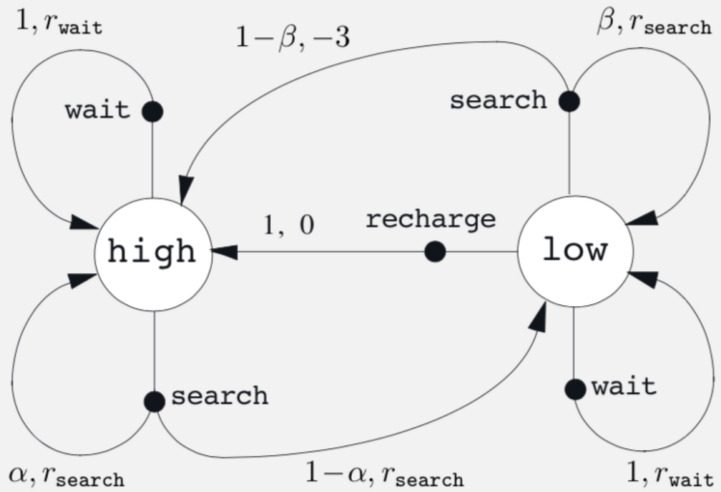
MDP Modelling: Recycling Robot

- A mobile robot has the job of collecting empty soda cans in an office
- It has sensors for detecting cans, and an arm and gripper that can pick them up and place them in an onboard bin
- It runs on a rechargeable battery
- Objective: How to search for cans without depleting the battery?
- How do we model this as an MDP?
  - What are the states and actions of the system?
- Modeling depends on the goal and convenience!

MDP Modelling: Recycling Robot

- States (of charge): {high, low}
- Actions: {search, wait, recharge}
- Reward for searching, waiting, and penalty for running out of charge

s	a	s'	$p(s' \mid s, a)$	$r(s, a, s')$
high	search	high	α	$r_{\texttt{search}}$
high	search	low	$1-\alpha$	$r_{\texttt{search}}$
low	search	high	$1-\beta$	-3
low	search	low	β	$r_{\texttt{search}}$
high	wait	high	1	$r_{\texttt{wait}}$
high	wait	low	0	-
low	wait	high	0	-
low	wait	low	1	$r_{\texttt{wait}}$
low	recharge	high	1	0
low	recharge	low	0	-



Control Policy

- A control policy π specifies the action a_t to take at each time step t

Control Policy

- A control policy π specifies the action a_t to take at each time step t
  - Control policy π can possibly select action a_t depending on the entire trajectory history $h_t = \{s_0, a_0, r_0, \ldots, s_{t-1}, a_{t-1}, r_{t-1}, s_t\}$

Control Policy

- A control policy π specifies the action a_t to take at each time step t
  - Control policy π can possibly select action a_t depending on the entire trajectory history $h_t = \{s_0, a_0, r_0, \ldots, s_{t-1}, a_{t-1}, r_{t-1}, s_t\}$
- Markov policy: A policy π is called a Markov policy if it selects action a_t based only on the current state s_t (not on the entire history h_t)

Control Policy

- A control policy π specifies the action a_t to take at each time step t
  - Control policy π can possibly select action a_t depending on the entire trajectory history $h_t = \{s_0, a_0, r_0, \ldots, s_{t-1}, a_{t-1}, r_{t-1}, s_t\}$
- Markov policy: A policy π is called a Markov policy if it selects action a_t based only on the current state s_t (not on the entire history h_t)
  - A Markov policy π can be specified as a sequence $\pi = (\bar{\pi}_0, \bar{\pi}_1, \ldots)$

Control Policy

- A control policy π specifies the action a_t to take at each time step t
  - Control policy π can possibly select action a_t depending on the entire trajectory history $h_t = \{s_0, a_0, r_0, \ldots, s_{t-1}, a_{t-1}, r_{t-1}, s_t\}$
- Markov policy: A policy π is called a Markov policy if it selects action a_t based only on the current state s_t (not on the entire history h_t)
  - A Markov policy π can be specified as a sequence $\pi = (\bar{\pi}_0, \bar{\pi}_1, \ldots)$
- Stationary policy: A Markov policy π is called a stationary policy if $\bar{\pi}_t = \bar{\pi}$ for all t, i.e., control law is the same for all time steps

Control Policy

- A control policy π specifies the action a_t to take at each time step t
  - Control policy π can possibly select action a_t depending on the entire trajectory history $h_t = \{s_0, a_0, r_0, \ldots, s_{t-1}, a_{t-1}, r_{t-1}, s_t\}$
- Markov policy: A policy π is called a Markov policy if it selects action a_t based only on the current state s_t (not on the entire history h_t)
  - A Markov policy π can be specified as a sequence $\pi = (\bar{\pi}_0, \bar{\pi}_1, \ldots)$
- Stationary policy: A Markov policy π is called a stationary policy if $\bar{\pi}_t = \bar{\pi}$ for all t, i.e., control law is the same for all time steps
- Control policy can be deterministic or stochastic

Control Policy

- A control policy π specifies the action a_t to take at each time step t
  - Control policy π can possibly select action a_t depending on the entire trajectory history $h_t = \{s_0, a_0, r_0, \ldots, s_{t-1}, a_{t-1}, r_{t-1}, s_t\}$
- Markov policy: A policy π is called a Markov policy if it selects action a_t based only on the current state s_t (not on the entire history h_t)
  - A Markov policy π can be specified as a sequence $\pi = (\bar{\pi}_0, \bar{\pi}_1, \ldots)$
- Stationary policy: A Markov policy π is called a stationary policy if $\bar{\pi}_t = \bar{\pi}$ for all t, i.e., control law is the same for all time steps
- Control policy can be deterministic or stochastic
- Stationary stochastic policy

Control Policy

- A control policy π specifies the action a_t to take at each time step t
  - Control policy π can possibly select action a_t depending on the entire trajectory history $h_t = \{s_0, a_0, r_0, \ldots, s_{t-1}, a_{t-1}, r_{t-1}, s_t\}$
- Markov policy: A policy π is called a Markov policy if it selects action a_t based only on the current state s_t (not on the entire history h_t)
  - A Markov policy π can be specified as a sequence $\pi = (\bar{\pi}_0, \bar{\pi}_1, \ldots)$
- Stationary policy: A Markov policy π is called a stationary policy if $\bar{\pi}_t = \bar{\pi}$ for all t, i.e., control law is the same for all time steps
- Control policy can be deterministic or stochastic
- Stationary stochastic policy
  - $\pi : \mathcal{S} \to \Delta(\mathcal{A})$

Control Policy

- A control policy π specifies the action a_t to take at each time step t
  - Control policy π can possibly select action a_t depending on the entire trajectory history $h_t = \{s_0, a_0, r_0, \ldots, s_{t-1}, a_{t-1}, r_{t-1}, s_t\}$
- Markov policy: A policy π is called a Markov policy if it selects action a_t based only on the current state s_t (not on the entire history h_t)
  - A Markov policy π can be specified as a sequence $\pi = (\bar{\pi}_0, \bar{\pi}_1, \ldots)$
- Stationary policy: A Markov policy π is called a stationary policy if $\bar{\pi}_t = \bar{\pi}$ for all t, i.e., control law is the same for all time steps
- Control policy can be deterministic or stochastic
- Stationary stochastic policy
  - $\pi : \mathcal{S} \to \Delta(\mathcal{A})$
  - $\pi(s, a)$ specifies the probability of taking control action a at state s

Control Policy

- A control policy π specifies the action a_t to take at each time step t
  - Control policy π can possibly select action a_t depending on the entire trajectory history $h_t = \{s_0, a_0, r_0, \ldots, s_{t-1}, a_{t-1}, r_{t-1}, s_t\}$
- Markov policy: A policy π is called a Markov policy if it selects action a_t based only on the current state s_t (not on the entire history h_t)
  - A Markov policy π can be specified as a sequence $\pi = (\bar{\pi}_0, \bar{\pi}_1, \ldots)$
- Stationary policy: A Markov policy π is called a stationary policy if $\bar{\pi}_t = \bar{\pi}$ for all t, i.e., control law is the same for all time steps
- Control policy can be deterministic or stochastic
- Stationary stochastic policy
  - $\pi : \mathcal{S} \to \Delta(\mathcal{A})$
  - $\pi(s, a)$ specifies the probability of taking control action a at state s
  - $a_t \sim \pi(s_t, \cdot)$

Control Policy

- A control policy π specifies the action a_t to take at each time step t
  - Control policy π can possibly select action a_t depending on the entire trajectory history $h_t = \{s_0, a_0, r_0, \ldots, s_{t-1}, a_{t-1}, r_{t-1}, s_t\}$
- Markov policy: A policy π is called a Markov policy if it selects action a_t based only on the current state s_t (not on the entire history h_t)
  - A Markov policy π can be specified as a sequence $\pi = (\bar{\pi}_0, \bar{\pi}_1, \ldots)$
- Stationary policy: A Markov policy π is called a stationary policy if $\bar{\pi}_t = \bar{\pi}$ for all t, i.e., control law is the same for all time steps
- Control policy can be deterministic or stochastic
- Stationary stochastic policy
  - $\pi : \mathcal{S} \to \Delta(\mathcal{A})$
  - $\pi(s, a)$ specifies the probability of taking control action a at state s
  - $a_t \sim \pi(s_t, \cdot)$
- Stationary deterministic policy

Control Policy

- A control policy π specifies the action a_t to take at each time step t
  - Control policy π can possibly select action a_t depending on the entire trajectory history $h_t = \{s_0, a_0, r_0, \ldots, s_{t-1}, a_{t-1}, r_{t-1}, s_t\}$
- Markov policy: A policy π is called a Markov policy if it selects action a_t based only on the current state s_t (not on the entire history h_t)
  - A Markov policy π can be specified as a sequence $\pi = (\bar{\pi}_0, \bar{\pi}_1, \ldots)$
- Stationary policy: A Markov policy π is called a stationary policy if $\bar{\pi}_t = \bar{\pi}$ for all t, i.e., control law is the same for all time steps
- Control policy can be deterministic or stochastic
- Stationary stochastic policy
  - $\pi : \mathcal{S} \to \Delta(\mathcal{A})$
  - $\pi(s, a)$ specifies the probability of taking control action a at state s
  - $a_t \sim \pi(s_t, \cdot)$
- Stationary deterministic policy
  - $\pi : \mathcal{S} \to \mathcal{A}$

Utility (Value) of Reward Sequence

- What preferences should an agent have over reward sequences?

Utility (Value) of Reward Sequence

- What preferences should an agent have over reward sequences?
  - More or less?: $\{1, 2, 2\}$ or $\{2, 3, 4\}$

Utility (Value) of Reward Sequence

- What preferences should an agent have over reward sequences?
  - More or less?: {1, 2, 2} or {2, 3, 4}
  - Now or later?: {0.1, 0.1, 10} or {10, 0.1, 0.1}

Utility (Value) of Reward Sequence

- What preferences should an agent have over reward sequences?
  - More or less?: $\{1, 2, 2\}$ or $\{2, 3, 4\}$
  - Now or later?: $\{0.1, 0.1, 10\}$ or $\{10, 0.1, 0.1\}$
- It is reasonable to maximize the sum of rewards

Utility (Value) of Reward Sequence

- What preferences should an agent have over reward sequences?
  - More or less?: $\{1, 2, 2\}$ or $\{2, 3, 4\}$
  - Now or later?: $\{0.1, 0.1, 10\}$ or $\{10, 0.1, 0.1\}$
- It is reasonable to maximize the sum of rewards
- It is also reasonable to prefer rewards now to rewards later

Utility (Value) of Reward Sequence

- What preferences should an agent have over reward sequences?
  - More or less?: {1, 2, 2} or {2, 3, 4}
  - Now or later?: {0.1, 0.1, 10} or {10, 0.1, 0.1}
- It is reasonable to maximize the sum of rewards
- It is also reasonable to prefer rewards now to rewards later
- One approach: value of rewards decay exponentially

$$\text{Value}(\{r_0, r_1, r_2, \ldots\}) = r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots$$

Utility (Value) of Reward Sequence

- What preferences should an agent have over reward sequences?
  - More or less?: {1, 2, 2} or {2, 3, 4}
  - Now or later?: {0.1, 0.1, 10} or {10, 0.1, 0.1}
- It is reasonable to maximize the sum of rewards
- It is also reasonable to prefer rewards now to rewards later
- One approach: value of rewards decay exponentially

$$\text{Value}(\{r_0, r_1, r_2, \ldots\}) = r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots$$

  - Discount factor $\gamma \in (0, 1)$

Utility (Value) of Reward Sequence

- What preferences should an agent have over reward sequences?
  - More or less?: {1, 2, 2} or {2, 3, 4}
  - Now or later?: {0.1, 0.1, 10} or {10, 0.1, 0.1}
- It is reasonable to maximize the sum of rewards
- It is also reasonable to prefer rewards now to rewards later
- One approach: value of rewards decay exponentially

$$\text{Value}(\{r_0, r_1, r_2, \ldots\}) = r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots$$

  - Discount factor $\gamma \in (0, 1)$
- Why discount?

Utility (Value) of Reward Sequence

- What preferences should an agent have over reward sequences?
  - More or less?: {1, 2, 2} or {2, 3, 4}
  - Now or later?: {0.1, 0.1, 10} or {10, 0.1, 0.1}
- It is reasonable to maximize the sum of rewards
- It is also reasonable to prefer rewards now to rewards later
- One approach: value of rewards decay exponentially

$$\text{Value}(\{r_0, r_1, r_2, \ldots\}) = r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots$$

  - Discount factor $\gamma \in (0, 1)$
- Why discount?
  - Mathematically convenient to discount rewards

Utility (Value) of Reward Sequence

- What preferences should an agent have over reward sequences?
  - More or less?: {1, 2, 2} or {2, 3, 4}
  - Now or later?: {0.1, 0.1, 10} or {10, 0.1, 0.1}
- It is reasonable to maximize the sum of rewards
- It is also reasonable to prefer rewards now to rewards later
- One approach: value of rewards decay exponentially

$$\text{Value}(\{r_0, r_1, r_2, \ldots\}) = r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots$$

  - Discount factor $\gamma \in (0, 1)$
- Why discount?
  - Mathematically convenient to discount rewards
  - Uncertainty about the future may not be fully represented

Utility (Value) of Reward Sequence

- What preferences should an agent have over reward sequences?
  - More or less?: {1, 2, 2} or {2, 3, 4}
  - Now or later?: {0.1, 0.1, 10} or {10, 0.1, 0.1}
- It is reasonable to maximize the sum of rewards
- It is also reasonable to prefer rewards now to rewards later
- One approach: value of rewards decay exponentially

$$\text{Value}(\{r_0, r_1, r_2, \ldots\}) = r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots$$

  - Discount factor $\gamma \in (0, 1)$
- Why discount?
  - Mathematically convenient to discount rewards
  - Uncertainty about the future may not be fully represented
  - If the reward is financial, immediate rewards may earn more interest than delayed rewards

Utility (Value) of Reward Sequence

- What preferences should an agent have over reward sequences?
  - More or less?: {1, 2, 2} or {2, 3, 4}
  - Now or later?: {0.1, 0.1, 10} or {10, 0.1, 0.1}
- It is reasonable to maximize the sum of rewards
- It is also reasonable to prefer rewards now to rewards later
- One approach: value of rewards decay exponentially

$$\text{Value}(\{r_0, r_1, r_2, \ldots\}) = r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots$$

  - Discount factor $\gamma \in (0, 1)$
- Why discount?
  - Mathematically convenient to discount rewards
  - Uncertainty about the future may not be fully represented
  - If the reward is financial, immediate rewards may earn more interest than delayed rewards
  - Animal/human behavior shows preference for immediate reward

Utility (Value) of Reward Sequence

- What preferences should an agent have over reward sequences?
  - More or less?: {1, 2, 2} or {2, 3, 4}
  - Now or later?: {0.1, 0.1, 10} or {10, 0.1, 0.1}
- It is reasonable to maximize the sum of rewards
- It is also reasonable to prefer rewards now to rewards later
- One approach: value of rewards decay exponentially

$$\text{Value}(\{r_0, r_1, r_2, \ldots\}) = r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots$$

  - Discount factor $\gamma \in (0, 1)$
- Why discount?
  - Mathematically convenient to discount rewards
  - Uncertainty about the future may not be fully represented
  - If the reward is financial, immediate rewards may earn more interest than delayed rewards
  - Animal/human behavior shows preference for immediate reward
  - It is sometimes possible to use undiscounted cumulative rewards processes (i.e., $\gamma = 1$), if all sequences terminate

Value of a Policy

- Value of a policy evaluated at state s is the expected cumulative (discounted) rewards obtained by taking action according that policy, starting from s

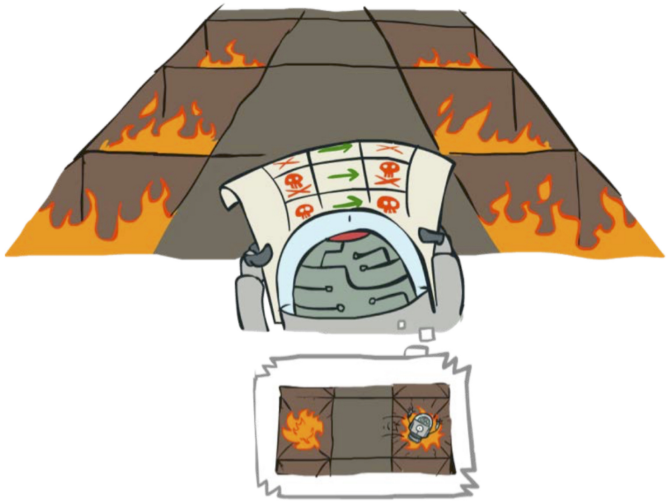
$$V_\pi(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 = s, \pi\right], \text{ where, } s_{t+1} \sim P(\cdot|s_t, a_t), \; a_t \sim \pi(h_t, \cdot)$$

Value of a Policy

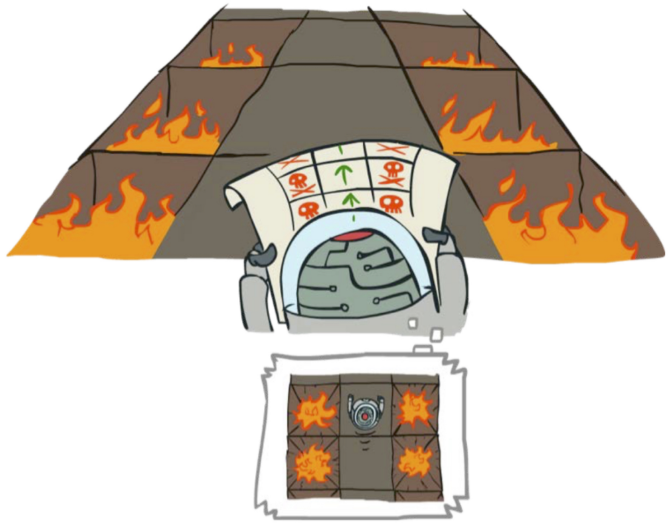
- Value of a policy evaluated at state s is the expected cumulative (discounted) rewards obtained by taking action according that policy, starting from s

$$V_\pi(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 = s, \pi\right], \text{ where, } s_{t+1} \sim P(\cdot|s_t, a_t),\ a_t \sim \pi(h_t, \cdot)$$

Always Go Right



Always Go Forward



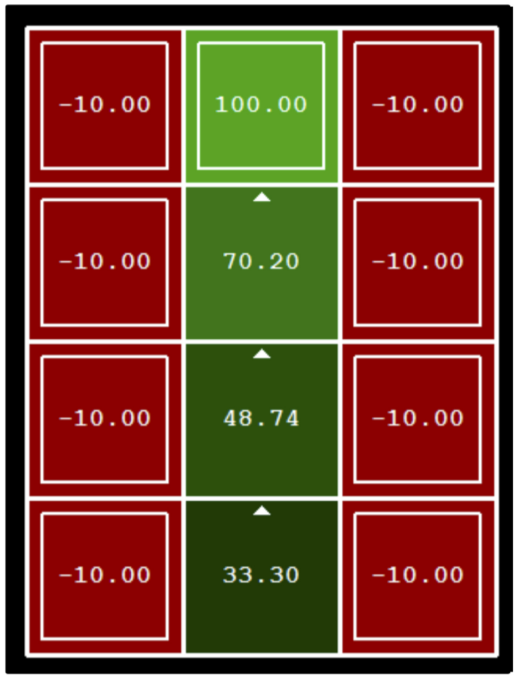
Value of a Policy

- Value of a policy evaluated at state x is the expected cumulative (discounted) rewards obtained by taking action according that policy, starting from x

$$V_\pi(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 = s, \pi\right], \text{ where, } s_{t+1} \sim P(\cdot|s_t, a_t), \ a_t \sim \pi(h_t, \cdot)$$

Always Go Right

Always Go Forward



Sufficiency of Stationary Deterministic Policy Class

- Goal: for each $s \in \mathcal{S}$, find a policy π that maximizes the value

$$\max_{\pi \in \Pi} \; V_\pi(s)$$

Sufficiency of Stationary Deterministic Policy Class

- Goal: for each $s \in \mathcal{S}$, find a policy π that maximizes the value

$$\max_{\pi \in \Pi} V_\pi(s)$$

- History dependent policy class is hard to handle due to exponentially large computation and storage requirement
- Can we focus only on stationary policies, without loss of optimality?

Sufficiency of Stationary Deterministic Policy Class

- Goal: for each $s \in \mathcal{S}$, find a policy π that maximizes the value

$$\max_{\pi \in \Pi} \; V_\pi(s)$$

- History dependent policy class is hard to handle due to exponentially large computation and storage requirement
- Can we focus only on stationary policies, without loss of optimality?

Theorem

For infinite horizon discounted reward MDP with finite state and action spaces, there exists a stationary and deterministic policy π^* such that, for all $s \in \mathcal{S}$,

$$V_{\pi^*}(s) = \max_{\pi \in \Pi} \; V_\pi(s)$$

Sufficiency of Stationary Deterministic Policy Class

- Goal: for each $s \in \mathcal{S}$, find a policy π that maximizes the value

$$\max_{\pi \in \Pi} \; V_\pi(s)$$

- History dependent policy class is hard to handle due to exponentially large computation and storage requirement
- Can we focus only on stationary policies, without loss of optimality?

Theorem

For infinite horizon discounted reward MDP with finite state and action spaces, there exists a stationary and deterministic policy π^* such that, for all $s \in \mathcal{S}$,

$$V_{\pi^*}(s) = \max_{\pi \in \Pi} \; V_\pi(s)$$

- **Reading assignment**: Proof of the above theorem, see [AJKS, Chapter 1], [BDP, Chapter 1] [Puterman, Chapter 5]
- For the rest of the semester, we will focus only on stationary policies

MDP Questions

MDP Questions

- How do we compute the value of a policy π?

$$V_\pi(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 = s, \pi\right], \text{ where, } s_{t+1} \sim P(\cdot|s_t, a_t), \; a_t \sim \pi(s_t, \cdot)$$

MDP Questions

- How do we compute the value of a policy π?

$$V_\pi(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 = s, \pi\right], \text{ where, } s_{t+1} \sim P(\cdot|s_t, a_t), \; a_t \sim \pi(s_t, \cdot)$$

- How do we compute the optimal policy?

$$\pi^\star = \arg\max_{\pi \in \Pi} \; V_\pi$$

MDP Questions

- How do we compute the value of a policy π?

$$V_\pi(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 = s, \pi\right], \text{ where, } s_{t+1} \sim P(\cdot|s_t, a_t), \; a_t \sim \pi(s_t, \cdot)$$

- How do we compute the optimal policy?

$$\pi^\star = \arg\max_{\pi \in \Pi} \; V_\pi$$

- Optimal value function

$$V^\star(s) = \max_{\pi \in \Pi} \; V_\pi(s)$$

MDP Questions

- How do we compute the value of a policy π?

$$V_\pi(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 = s, \pi\right], \text{ where, } s_{t+1} \sim P(\cdot|s_t, a_t),\ a_t \sim \pi(s_t, \cdot)$$

- How do we compute the optimal policy?

$$\pi^\star = \arg\max_{\pi \in \Pi} V_\pi$$

- Optimal value function

$$V^\star(s) = \max_{\pi \in \Pi} V_\pi(s)$$

  - By definition, $V^\star(s) = V_{\pi^\star}(s)$

Review: Norm of a Vector

Norm of a Vector

- Let $u = (u_1, \ldots, u_n) \in \mathbb{R}^n$

Norm of a Vector

- Let $u = (u_1, \ldots, u_n) \in \mathbb{R}^n$
- l_p norm of u for $p \geq 1$ is defined as

$$\|u\|_p = \left(\sum_{i=1}^{n} |u_i|^p \right)^{1/p}$$

Norm of a Vector

- Let $u = (u_1, \ldots, u_n) \in \mathbb{R}^n$
- l_p norm of u for $p \geq 1$ is defined as

$$\|u\|_p = \left(\sum_{i=1}^{n} |u_i|^p \right)^{1/p}$$

- Euclidean norm (l_2 norm) : $\|u\|_2 = \left(\sum_{i=1}^{n} (u_i)^2 \right)^{1/2}$

Norm of a Vector

- Let $u = (u_1, \ldots, u_n) \in \mathbb{R}^n$
- l_p norm of u for $p \geq 1$ is defined as

$$\|u\|_p = \left(\sum_{i=1}^{n} |u_i|^p \right)^{1/p}$$

- Euclidean norm (l_2 norm) : $\|u\|_2 = \left(\sum_{i=1}^{n} (u_i)^2 \right)^{1/2}$
- l_1 norm: $\|u\|_1 = \sum_{i=1}^{n} |u_i|$

Norm of a Vector

- Let $u = (u_1, \ldots, u_n) \in \mathbb{R}^n$
- l_p norm of u for $p \geq 1$ is defined as

$$\|u\|_p = \left(\sum_{i=1}^{n} |u_i|^p \right)^{1/p}$$

- Euclidean norm (l_2 norm) : $\|u\|_2 = \left(\sum_{i=1}^{n} (u_i)^2\right)^{1/2}$
- l_1 norm: $\|u\|_1 = \sum_{i=1}^{n} |u_i|$
- Maximum/supremum norm (l_∞ norm): $\|u\|_\infty = \max_i |u_i|$