CS 747, Autumn 2023: Lecture 12

Shivaram Kalyanakrishnan

Department of Computer Science and Engineering
Indian Institute of Technology Bombay

Autumn 2023
1. Reinforcement learning problem

2. Upcoming topics

3. Applications
Reinforcement Learning

1. Reinforcement learning problem

2. Upcoming topics

3. Applications
The Learning Setting

Underlying MDP:

$$s_1 \rightarrow 0.5, 0 \rightarrow 0.25, -1 \rightarrow 1, 1 \rightarrow 0.5, -1 \rightarrow 1, 2 \rightarrow 0.75, -2 \rightarrow 0.5, 3 \rightarrow 0.5, 3 \rightarrow s_3$$

$$s_2 \rightarrow 0.5, -1 \rightarrow 1, 1 \rightarrow 1, 2 \rightarrow 0.75, -2 \rightarrow 0.5, 3 \rightarrow 0.5, 3 \rightarrow s_3$$

$$s_3 \rightarrow 0.5, 3$$

$\gamma = 0.9$
The Learning Setting

Underlying MDP:

Agent's view:

From current state, agent takes action.
Environment (MDP) decides next state and reward.
Possible history:

History conveys information about the MDP to the agent.
Can the agent eventually take optimal actions?
The Learning Setting

Underlying MDP:

\[ s_1 \overset{1,1}{\rightarrow} s_2 \overset{1,2}{\rightarrow} s_3 \overset{0.5, -1}{\rightarrow} s_1 \]

\[ s_1 \overset{0.5, 3}{\rightarrow} s_3 \]

\[ s_2 \overset{0.75, -2}{\rightarrow} s_3 \]

\[ s_3 \overset{0.25, -1}{\rightarrow} s_1 \]

\[ \gamma = 0.9 \]

Agent’s view:

\[ s_1 \overset{\gamma = 0.9}{\rightarrow} s_3 \]

\[ s_2 \]

From current state, agent takes action.
The Learning Setting

Underlying MDP:

Agent’s view:

- From current state, agent takes action.
- Environment (MDP) decides next state and reward.
The Learning Setting

Underlying MDP:

Agent's view:

- From current state, agent takes action.
- Environment (MDP) decides next state and reward.
- Possible history: $s_2$, RED, $-2$, $s_3$, BLUE, $1$, $s_1$, RED, $0$, $s_1$, ...
From current state, agent takes action.
Environment (MDP) decides next state and reward.
Possible history: $s_2$, RED, $-2$, $s_3$, BLUE, 1, $s_1$, RED, 0, $s_1$, . . .
History conveys information about the MDP to the agent.
The Learning Setting

Underlying MDP:

Agent's view:

From current state, agent takes action.
Environment (MDP) decides next state and reward.
Possible history: \(s_2, \text{RED}, -2, s_3, \text{BLUE}, 1, s_1, \text{RED}, 0, s_1, \ldots\)
History conveys information about the MDP to the agent.
Can the agent eventually take optimal actions?
Planning and Learning

In the planning setting, the entire MDP \((S, A, T, R, \gamma)\) is available as an input. Obtaining \(\pi^*\) is a **computational** problem.
Planning and Learning

- In the planning setting, the entire MDP \((S, A, T, R, \gamma)\) is available as an input. Obtaining \(\pi^*\) is a computational problem.

- In the learning setting, the agent only knows \(S, A, \gamma\), and sometimes \(R\). It has to make inferences about \(T\) (and sometimes \(R\)) by taking actions from different states.
Planning and Learning

- In the planning setting, the entire MDP \((S, A, T, R, \gamma)\) is available as an input. Obtaining \(\pi^*\) is a **computational** problem.

- In the learning setting, the agent only knows \(S, A, \gamma\), and sometimes \(R\). It has to make inferences about \(T\) (and sometimes \(R\)) by taking actions from different states.

- For \(t \geq 0\), let \(h^t = (s^0, a^0, r^0, s^1, a^1, r^1, s^2, \ldots, s^t)\) denote a \(t\)-length **history**.
Planning and Learning

- In the planning setting, the entire MDP \((S, A, T, R, \gamma)\) is available as an input. Obtaining \(\pi^*\) is a computational problem.

- In the learning setting, the agent only knows \(S, A, \gamma\), and sometimes \(R\). It has to make inferences about \(T\) (and sometimes \(R\)) by taking actions from different states.

- For \(t \geq 0\), let \(h^t = (s^0, a^0, r^0, s^1, a^1, r^1, s^2, \ldots, s^t)\) denote a \(t\)-length history.

- A learning algorithm \(L\) is a mapping from the set of all histories to the set of all (probability distributions over) actions.
Planning and Learning

- In the planning setting, the entire MDP \((S, A, T, R, \gamma)\) is available as an input. Obtaining \(\pi^*\) is a computational problem.

- In the learning setting, the agent only knows \(S, A, \gamma,\) and sometimes \(R\). It has to make inferences about \(T\) (and sometimes \(R\)) by taking actions from different states.

- For \(t \geq 0\), let \(h_t = (s^0, a^0, r^0, s^1, a^1, r^1, s^2, \ldots, s^t)\) denote a \(t\)-length history.

- A **learning algorithm** \(L\) is a mapping from the set of all histories to the set of all (probability distributions over) actions.

- **Learning problem**: Can we construct \(L\) such that

\[
\lim_{H \to \infty} \frac{1}{H} \left( \sum_{t=0}^{H-1} \mathbb{P}\{a^t \sim L(h^t) \text{ is an optimal action for } s^t\} \right) = 1?
\]
1. Reinforcement learning problem

2. Upcoming topics

3. Applications
Upcoming Topics

- Temporal difference learning: prediction and control
  - On-line estimation of value function/action value function.

- Generalisation and function approximation
  - Compact representations to handle large state spaces.

- Policy gradient and policy search methods
  - Direct search over policy parameters.

- Model-based RL
  - Using (approximate) representations of $T$ and $R$ for learning.

- Batch RL
  - Storing and learning from a sequence of transitions (batch).

- Monte Carlo tree search
  - Planning for action selection.

- Multiagent RL
  - Coping with other learning agents.

- Applications
  - ATARI games (Mnih et al. (2015)), Go (Silver et al. (2016)).
Upcoming Topics

- Temporal difference learning: prediction and control
  - On-line estimation of value function/action value function.

- Generalisation and function approximation
  - Compact representations to handle large state spaces.
Upcoming Topics

- Temporal difference learning: prediction and control
  - On-line estimation of value function/action value function.

- Generalisation and function approximation
  - Compact representations to handle large state spaces.

- Policy gradient and policy search methods
  - Direct search over policy parameters.
Upcoming Topics

- **Temporal difference learning: prediction and control**
  - On-line estimation of value function/action value function.
- **Generalisation and function approximation**
  - Compact representations to handle large state spaces.
- **Policy gradient and policy search methods**
  - Direct search over policy parameters.
- **Model-based RL**
  - Using (approximate) representations of $T$ and $R$ for learning.
Upcoming Topics

- Temporal difference learning: prediction and control
  - On-line estimation of value function/action value function.

- Generalisation and function approximation
  - Compact representations to handle large state spaces.

- Policy gradient and policy search methods
  - Direct search over policy parameters.

- Model-based RL
  - Using (approximate) representations of $T$ and $R$ for learning.

- Batch RL
  - Storing and learning from a sequence of transitions (batch).
Upcoming Topics

- Temporal difference learning: prediction and control
  - On-line estimation of value function/action value function.

- Generalisation and function approximation
  - Compact representations to handle large state spaces.

- Policy gradient and policy search methods
  - Direct search over policy parameters.

- Model-based RL
  - Using (approximate) representations of $T$ and $R$ for learning.

- Batch RL
  - Storing and learning from a sequence of transitions (batch).

- Monte Carlo tree search
  - Planning for action selection.
Upcoming Topics

- Temporal difference learning: prediction and control
  - On-line estimation of value function/action value function.
- Generalisation and function approximation
  - Compact representations to handle large state spaces.
- Policy gradient and policy search methods
  - Direct search over policy parameters.
- Model-based RL
  - Using (approximate) representations of $T$ and $R$ for learning.
- Batch RL
  - Storing and learning from a sequence of transitions (batch).
- Monte Carlo tree search
  - Planning for action selection.
- Multiagent RL
  - Coping with other learning agents.
Upcoming Topics

- Temporal difference learning: prediction and control
  - On-line estimation of value function/action value function.
- Generalisation and function approximation
  - Compact representations to handle large state spaces.
- Policy gradient and policy search methods
  - Direct search over policy parameters.
- Model-based RL
  - Using (approximate) representations of $T$ and $R$ for learning.
- Batch RL
  - Storing and learning from a sequence of transitions (batch).
- Monte Carlo tree search
  - Planning for action selection.
- Multiagent RL
  - Coping with other learning agents.
- Applications
  - ATARI games (Mnih et al. (2015)), Go (Silver et al. (2016)).
Reinforcement Learning

1. Reinforcement learning problem

2. Upcoming topics

3. Applications
Board Games

Backgammon

Go

Chess


Robotics and Control

Reference: Ng et al. (2003).

Video Games


Computer Systems

Optimising a memory controller

Reference: İpek et al. (2008).

Healthcare

Adaptive treatment of epilepsy

Reference: Guez et al. (2008).

Finance

Stock trading

Reinforcement Learning

1. Reinforcement learning problem

2. Upcoming topics

3. Applications