Reinforcement Learning
An Introduction/Overview

Dennis Wagner
Humboldt University of Berlin
Overview

Reinforcement Learning

1 Introduction
Overview

Reinforcement Learning

1. Introduction

2. Markov Decision Processes
Overview

Reinforcement Learning

1 Introduction

2 Markov Decision Processes

3 Generalized Policy Iteration
Overview

Reinforcement Learning

1. Introduction
2. Markov Decision Processes
3. Generalized Policy Iteration
4. Function Approximation
Overview

Reinforcement Learning

1. Introduction
2. Markov Decision Processes
3. Generalized Policy Iteration
4. Function Approximation
5. An Example
Introduction
Examples of Reinforcement Learning Problems

**Industrial Robotics**

Industrial robots learning to manipulate objects.

Examples of Reinforcement Learning Problems

Games

Teaching a computer program to play complex games.

Examples of Reinforcement Learning Problems

Games

Teaching a computer program to play complex games.

Examples of Reinforcement Learning Problems

Recommender Systems

Modelling interactions with the user as an MDP.

The general setting
Markov Decision Processes
Running example - Gridworld
Elements of the RL Model - Gridworld
Elements of the RL Model - Gridworld
Elements of the RL Model - Gridworld
Elements of the RL Model - MDP

\[ s \xrightarrow{a} p(s'|s,a) \xrightarrow{r(s,a,s')} s' \]
Elements of the RL Model - policy
Elements of the RL Model - policy
Example: Gridworld
Example: Gridworld

<table>
<thead>
<tr>
<th></th>
<th>←</th>
<th>←</th>
<th>↓</th>
<th>←</th>
</tr>
</thead>
<tbody>
<tr>
<td>↑</td>
<td>→</td>
<td>↑</td>
<td>↓</td>
<td>←</td>
</tr>
<tr>
<td>↑</td>
<td>→</td>
<td>→</td>
<td>↑</td>
<td>↑</td>
</tr>
<tr>
<td>↑</td>
<td>↑</td>
<td>→</td>
<td>→</td>
<td>↑</td>
</tr>
<tr>
<td>↑</td>
<td>↑</td>
<td>→</td>
<td>→</td>
<td>↑</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>0.8</th>
<th>-1.2</th>
<th>-1.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.4</td>
<td>0.6</td>
<td>-1</td>
<td>-1.2</td>
</tr>
<tr>
<td>0.8</td>
<td>-1.2</td>
<td>-1</td>
<td>-1.4</td>
<td></td>
</tr>
<tr>
<td>0.6</td>
<td>-1.4</td>
<td>-2</td>
<td>-1.8</td>
<td>-1.6</td>
</tr>
<tr>
<td>?</td>
<td>-1.6</td>
<td>-2.2</td>
<td>-2</td>
<td>-1.8</td>
</tr>
</tbody>
</table>
Example: Gridworld

$$V^\pi(s) =$$

<table>
<thead>
<tr>
<th></th>
<th>←</th>
<th>←</th>
<th>↓</th>
<th>←</th>
</tr>
</thead>
<tbody>
<tr>
<td>↑</td>
<td>→</td>
<td>↑</td>
<td>↓</td>
<td>←</td>
</tr>
<tr>
<td>↑</td>
<td>→</td>
<td>→</td>
<td>↑</td>
<td>↑</td>
</tr>
<tr>
<td>↑</td>
<td>↑</td>
<td>→</td>
<td>→</td>
<td>↑</td>
</tr>
<tr>
<td>↑</td>
<td>↑</td>
<td>→</td>
<td>→</td>
<td>↑</td>
</tr>
</tbody>
</table>

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>0.8</th>
<th>-1.2</th>
<th>-1.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.4</td>
<td>0.6</td>
<td>-1</td>
<td>-1.2</td>
</tr>
<tr>
<td>0.8</td>
<td>-1.2</td>
<td>-1</td>
<td>-1.4</td>
<td></td>
</tr>
<tr>
<td>0.6</td>
<td>-1.4</td>
<td>-2</td>
<td>-1.8</td>
<td>-1.6</td>
</tr>
<tr>
<td>?</td>
<td>-1.6</td>
<td>-2.2</td>
<td>-2</td>
<td>-1.8</td>
</tr>
</tbody>
</table>
Example: Gridworld

\[ V^\pi(s) = -0.2 \]
Example: Gridworld

\[ V^\pi(s) = -0.2 - 0.2 \]
Example: Gridworld

\[ V^\pi(s) = -0.2 - 0.2 - 0.2 \]
Example: Gridworld

\[
V^\pi(s) = -0.2 - 0.2 - 0.2 + 1
\]
Example: Gridworld

<table>
<thead>
<tr>
<th></th>
<th>←</th>
<th>←</th>
<th>↓</th>
<th>←</th>
</tr>
</thead>
<tbody>
<tr>
<td>↑</td>
<td>→</td>
<td>↑</td>
<td>↓</td>
<td>←</td>
</tr>
<tr>
<td>↑</td>
<td>→</td>
<td>→</td>
<td>↑</td>
<td>↑</td>
</tr>
<tr>
<td>↑</td>
<td>↑</td>
<td>→</td>
<td>→</td>
<td>↑</td>
</tr>
<tr>
<td>↑</td>
<td>↑</td>
<td>→</td>
<td>→</td>
<td>↑</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>0.8</th>
<th>-1.2</th>
<th>-1.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.4</td>
<td>0.6</td>
<td>-1</td>
<td>-1.2</td>
</tr>
<tr>
<td>0.8</td>
<td>-1.2</td>
<td>-1</td>
<td>-1.4</td>
<td></td>
</tr>
<tr>
<td>0.6</td>
<td>-1.4</td>
<td>-2</td>
<td>-1.8</td>
<td>-1.6</td>
</tr>
<tr>
<td>0.4</td>
<td>-1.6</td>
<td>-2.2</td>
<td>-2</td>
<td>-1.8</td>
</tr>
</tbody>
</table>
Example: Gridworld (2)

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>0.8</th>
<th>0.6</th>
<th>0.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.8</td>
<td>0.6</td>
<td>0.4</td>
<td>0.2</td>
</tr>
<tr>
<td>0.8</td>
<td>0.6</td>
<td>0.4</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>0.6</td>
<td>0.4</td>
<td>0.2</td>
<td></td>
<td>-0.2</td>
</tr>
<tr>
<td>0.4</td>
<td>0.2</td>
<td></td>
<td>-0.2</td>
<td>-0.4</td>
</tr>
</tbody>
</table>
Example: Gridworld (2)

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>0.8</th>
<th>0.6</th>
<th>0.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.8</td>
<td>0.6</td>
<td>0.4</td>
<td>0.2</td>
</tr>
<tr>
<td>0.8</td>
<td>0.6</td>
<td>0.4</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0.6</td>
<td>0.4</td>
<td>0.2</td>
<td>0</td>
<td>-0.2</td>
</tr>
<tr>
<td>0.4</td>
<td>0.2</td>
<td>0</td>
<td>-0.2</td>
<td>-0.4</td>
</tr>
</tbody>
</table>

Left arrow represents the agent's movement.
Example: Gridworld

```
  ▶  ▶  ▼  
  ◀  ▼  ◀  
  ▶  ▼  ◀  
```

```
-0.3  0.6  0.7  
-1  -1.1  0.8  -1  
0.9  +1  0.9  
```
Example: Gridworld

- Gridworld example with arrows and values.
- The grid is divided into smaller sections with arrows indicating movement directions.
- Values are shown in each block, indicating rewards or penalties for moving.
- The goal is to navigate the grid to reach the green square with a +1 reward.
Bellman Theorem
Generalized Policy Iteration
Example: Gridworld
Example: Gridworld

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image_url" alt="Diagram" /></td>
<td><img src="image_url" alt="Diagram" /></td>
<td><img src="image_url" alt="Diagram" /></td>
<td><img src="image_url" alt="Diagram" /></td>
</tr>
</tbody>
</table>

- States
- Actions
- Rewards

-0.3 0.6 0.7
-1 -1.1 0.8 -1
0.9 +1 0.9
0.5 0.6 0.7
-1 0.8 0.8 -1
0.9 +1 0.9
Generalized Policy Iteration

1: function GPI
2: \[ \pi \leftarrow \text{arbitrary policy} \]

7: return \( \pi \)
8: end function
function GPI

π ← arbitrary policy
repeat

V ← Vπ  
Policy Evaluation
π ← greedy (V)  
Policy Improvement
until convergence
return π
end function
Generalized Policy Iteration

1: function GPI
2: \[ \pi \leftarrow \text{arbitrary policy} \]
3: repeat
4: \[ V \leftarrow V^\pi \] ▶ Policy Evaluation
5: until convergence
6: return \( \pi \)
7: end function
Generalized Policy Iteration

1: function GPI
2: \( \pi \leftarrow \text{arbitrary policy} \)
3: repeat
4: \( V \leftarrow V^\pi \) ▷ Policy Evaluation
5: \( \pi \leftarrow \text{greedy}(V) \) ▷ Policy Improvement
6: until convergence
7: return \( \pi \)
8: end function
Offpolicy (TD) learning

\[ a^* = \arg \max_{a} Q(s,a) \]

\[ a_{t+1} = \arg \max_{a} \pi(s,a) \]
Function Approximation
Example: Q
Example: Q
Example: Q
Experience Replay

Reinforcement Learning — An Introduction/Overview

Memory

\[
\begin{array}{cccc}
  s_0 & a_0 & r_1 & s_1 \\
  s_1 & a_1 & r_2 & s_2 \\
  \cdots \\
  s_t & a_t & r_{t+1} & s_{t+1} \\
\end{array}
\]
Fixed target nets

\[ S, a \rightarrow Q \]
Fixed target nets

\[ (s, a) \rightarrow \text{eval} \rightarrow Q \]

\[ (s, a) \rightarrow \text{target} \rightarrow Q \]

The diagram shows two networks, an evaluation network (eval) and a target network, connected by a copy mechanism. The input \((s, a)\) is fed into both networks, which then output their respective Q-values.
Fixed target nets

\[ s, a \]

\[ \text{eval} \]

\[ Q \]

update parameters

copy

\[ s, a \]

\[ \text{target} \]

\[ Q \]

compute targets
Actor Critic methods

Agent

Environment

state
$s_t$

reward
$r_t$

$r_{t+1}$

action
$a_t$

$s_{t+1}$
Actor Critic methods
An Example
Monte Carlo Tree Search

Selection
Monte Carlo Tree Search

Selection

Expansion
Monte Carlo Tree Search

Selection

Expansion

Evaluation
Monte Carlo Tree Search
AlphaGO

![Diagram of AlphaGO process](image)

1. **Policy** $p_\sigma$
   - 13 layer NN
   - **self-play**

2. **Policy** $p_\rho$

3. **Policy** $p_\Pi$

**Notes:**
- REINFORCE + Baseline ($V_\Pi$)
AlphaGO

Reinforcement Learning — An Introduction/Overview
Selection

\[ \text{max } Q + u(P) \]
AlphaGO - MCTS

Selection

\[
\text{max } Q + u(P)
\]

Expansion

\[
p_\rho(\cdot | P)
\]
AlphaGO - MCTS

Selection

Expansion

max \( Q + u(P) \)

Evaluation

\( p_\rho(\cdot | P) \)

\( V_W(\cdot) \)

\( p_\pi \)

\( r(\cdot) \)
AlphaGO - MCTS

**Selection**
\[
\max Q + u(P) \quad p_\rho(\cdot | P)
\]

**Expansion**

**Evaluation**
\[
V_W(\cdot) \quad p_\pi \quad r(\cdot)
\]

**Backup**
\[
Q \quad V_W(\cdot) \quad r(\cdot)
\]
Thank you.

Any Questions?
Exercises and solutions to accompany sutton’s book and david silver’s course.  


Ucl course on rl.  


References II


Further references

- Nuts and Bolts of Deep RL Experimentation
  (https://www.youtube.com/watch?v=8EcdaCk9KaQ)
- Trust Region Policy Optimization
- Proximal Policy Optimization Algorithms
  (https://blog.openai.com/openai-baselines-ppo/)
- OpenAI 5 (DOTA2)
  (https://blog.openai.com/openai-five-benchmark-results/)
- Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks
- DQN extensions