Introduction and Basics of Markov Decision Process

Wen Sun
CS 6789: Foundations of Reinforcement Learning
Progress of RL in Practice

TD GAMMON [Tesauro 95]

[AlphaZero, Silver et.al, 17]

[OpenAI Five, 18]
RL in Real World:
RL in Real World:
RL in Real World:
RL in Real World:

Training Language models using RL, e.g., chatGPT

```
USER: this code is not working like I expect — how do I fix it?

resultWorkerErr := make(chan error)
def err, close(resultWorkerErr)

f func() {
    defer cancel()
    resultWorkerErr < b.resultWorker(ctx)
}

err := b.error(ctx)
cancel()
if err = nil {
    return <resultWorkerErr
}
return multierror.Append(err, <resultWorkerErr)

CHATGPT: It's difficult to say what's wrong with the code without more context. Can you provide more information about what the code is supposed to do and what isn't working as expected? Also, is this the entire code or just a part of it?

USER: just part of it — the error never surfaces. I think there is something wrong with the channel

CHATGPT: It's difficult to say without more information about what the code is supposed to do and what's happening when it's executed. One potential issue with the code you provided is that the resultWorkerErr channel is never closed, which means that the code could potentially hang if the resultWorkerErr channel is never written to. This
This course focuses on **RL Theory**

When and Why RL works!
(Convergence, sample / computation complexity, etc)
Four main themes we will cover in this course:

1. Fundamentals (MDPs and Optimal planning)
2. Exploration (sample complexity)
3. Policy Gradient (global convergence)
4. Learning Partially observable models
Logistics

Four (HW0-HW3) assignments (total 55%), Course Project (40%), Reading (5%)

(HW0 10%, HW1-3 15% each)

HW0 is out today and due in one week
Prerequisites (HW0)

Deep understanding of Machine Learning, Optimization, Statistics

ML: sample complexity analysis for supervised learning (PAC)

Opt: Convex (linear) optimization, e.g., gradient decent for convex functions

Stats: basics of concentration (e.g., Hoeffding’s), tricks such as union bound
Prerequisites (HW0)

Deep understanding of Machine Learning, Optimization, Statistics

ML: sample complexity analysis for supervised learning (PAC)

Opt: Convex (linear) optimization, e.g., gradient decent for convex functions

Stats: basics of concentration (e.g., Hoeffding’s), tricks such as union bound

Check out HW0 asap!
Course projects (40%)

- Team work: size 3
- Midterm report (5%), Final presentation (15%), and Final report (20%)

- Basics: **survey** of a set of similar RL theory papers. Reproduce analysis and provide a coherent story
- Advanced: **identify** extensions of existing RL papers, **formulate** theory questions, and **provide** proofs
Course Notes:
Reinforcement Learning Theory & Algorithms

• Book website: https://rltheorybook.github.io/

• Many lectures will correspond to chapters in Version 2.
  • Reading assignment (5%) is from this book and additional notes

• Please let us know if you find typos/errors in the book!
  We appreciate it!
Outline

1. Definition of infinite horizon discounted MDPs

2. Bellman Optimality

3. State-action distribution
Supervised Learning
Supervised Learning

Given i.i.d examples at training:
Supervised Learning

Given i.i.d examples at training:

\[
(f, \text{cat}) \quad (\text{cat}) \quad (\text{dog})
\]

\[f \in \mathcal{F}\]
Supervised Learning

Given i.i.d examples at training:

\[
(f \in \mathcal{F}, \text{cat}) (f \in \mathcal{F}, \text{cat}) (f \in \mathcal{F}, \text{dog})
\]

Passive:

Prediction

Data Distribution
Agent Linear
Selected Actions:

RIGHT

SPEED

Active: Decisions ➔ Data Distribution
Active: Decisions ➔ Data Distribution
Markov Decision Process

Learning Agent

\[ a \sim \pi(s) \]

Policy: determine action based on state

Environment
Markov Decision Process

Policy: determine action based on state

$a \sim \pi(s)$

Send reward and next state from a Markovian transition dynamics

$r(s, a), s' \sim P(\cdot | s, a)$
Markov Decision Process

Policy: determine action based on state

Multiple Steps

Send reward and next state from a Markovian transition dynamics

\[
r(s, a), s' \sim P( \cdot | s, a)
\]
**Markov Decision Process**

**Policy**: determine action based on state

\[ a \sim \pi(s) \]

Multiple Steps

Send **reward** and **next state** from a Markovian transition dynamics

\[ r(s, a), s' \sim P( \cdot \mid s, a) \]
Markov Decision Process

Policy: determine action based on state

\[ a \sim \pi(s) \]

Send reward and next state from a Markovian transition dynamics

\[ r(s, a), s' \sim P(\cdot | s, a) \]
Markov Decision Process

Policy: determine action based on state

Multiple Steps

Send reward and next state from a Markovian transition dynamics

\[ a \sim \pi(s) \]

\[ r(s, a), s' \sim P( \cdot \mid s, a) \]

\[ s_0 \sim \mu_0, a_0 \sim \pi(s_0), r_0, s_1 \sim P(s_0, a_0), a_1 \sim \pi(s_1), r_1 \ldots \]
<table>
<thead>
<tr>
<th></th>
<th>Learn from Experience</th>
<th>Generalize</th>
<th>Interactive</th>
<th>Exploration</th>
<th>Credit assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supervised Learning</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reinforcement Learning</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Learn from Experience</td>
<td>Generalize</td>
<td>Interactive</td>
<td>Exploration</td>
<td>Credit assignment</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------------------</td>
<td>------------</td>
<td>-------------</td>
<td>-------------</td>
<td>-------------------</td>
</tr>
<tr>
<td><strong>Supervised Learning</strong></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Reinforcement Learning</strong></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table content based on slides from Emma Brunskill
<table>
<thead>
<tr>
<th></th>
<th>Learn from Experience</th>
<th>Generalize</th>
<th>Interactive</th>
<th>Exploration</th>
<th>Credit assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Supervised Learning</strong></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Reinforcement Learning</strong></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Learn from Experience</td>
<td>Generalize</td>
<td>Interactive</td>
<td>Exploration</td>
<td>Credit assignment</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-----------------------</td>
<td>------------</td>
<td>-------------</td>
<td>-------------</td>
<td>-------------------</td>
</tr>
<tr>
<td><strong>Supervised Learning</strong></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Reinforcement Learning</strong></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

Table content based on slides from Emma Brunskill
<table>
<thead>
<tr>
<th></th>
<th>Learn from Experience</th>
<th>Generalize</th>
<th>Interactive</th>
<th>Exploration</th>
<th>Credit assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Supervised Learning</strong></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Reinforcement Learning</strong></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Learn from Experience</td>
<td>Generalize</td>
<td>Interactive</td>
<td>Exploration</td>
<td>Credit assignment</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------</td>
<td>------------</td>
<td>-------------</td>
<td>-------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>Supervised Learning</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reinforcement Learning</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Infinite horizon Discounted MDP

\[ M = \{ S, A, P, r, \mu_0, \gamma \} \]

\[ P : S \times A \mapsto \Delta(S), \quad r : S \times A \rightarrow [0,1], \quad \gamma \in [0,1) \]
Infinite horizon Discounted MDP

\[ \mathcal{M} = \{ S, A, P, r, \mu_0, \gamma \} \]

\[ P : S \times A \mapsto \Delta(S), \quad r : S \times A \to [0,1], \quad \gamma \in [0,1) \]

Policy \( \pi : S \mapsto \Delta(A) \)
Infinite horizon Discounted MDP

\[ \mathcal{M} = \{S, A, P, r, \mu_0, \gamma\} \]

\[ P : S \times A \mapsto \Delta(S), \quad r : S \times A \to [0,1], \quad \gamma \in [0,1) \]

Policy \( \pi : S \mapsto \Delta(A) \)

Value function \( V^\pi(s) = \mathbb{E} \left[ \sum_{h=0}^{\infty} \gamma^h r(s_h, a_h) \mid s_0 = s, a_h \sim \pi(s_h), s_{h+1} \sim P(\cdot \mid s_h, a_h) \right] \)
Infinite horizon Discounted MDP

\[ \mathcal{M} = \{ S, A, P, r, \mu_0, \gamma \} \]

\[ P : S \times A \mapsto \Delta(S), \quad r : S \times A \to [0,1], \quad \gamma \in [0,1) \]

Policy \( \pi : S \mapsto \Delta(A) \)

Value function
\[ V^\pi(s) = \mathbb{E} \left[ \sum_{h=0}^{\infty} \gamma^h r(s_h, a_h) \bigg| s_0 = s, a_h \sim \pi(s_h), s_{h+1} \sim P(\cdot | s_h, a_h) \right] \]

Q function
\[ Q^\pi(s, a) = \mathbb{E} \left[ \sum_{h=0}^{\infty} \gamma^h r(s_h, a_h) \bigg| (s_0, a_0) = (s, a), a_h \sim \pi(s_h), s_{h+1} \sim P(\cdot | s_h, a_h) \right] \]
Bellman Equation:

\[ V^\pi(s) = \mathbb{E} \left[ \sum_{h=0}^{\infty} \gamma^h r(s_h, a_h) \mid s_0 = s, a_h \sim \pi(s_h), s_{h+1} \sim P(\cdot \mid s_h, a_h) \right] \]
Bellman Equation:

\[
V^\pi(s) = \mathbb{E} \left[ \sum_{h=0}^{\infty} \gamma^h r(s_h, a_h) \mid s_0 = s, a_h \sim \pi(s_h), s_{h+1} \sim P(\cdot \mid s_h, a_h) \right]
\]

\[
V^\pi(s) = \mathbb{E}_{a \sim \pi(s)} \left[ r(s, a) + \gamma \mathbb{E}_{s' \sim P(\cdot \mid s, a)} V^\pi(s') \right]
\]
Bellman Equation:

\[
V^\pi(s) = \mathbb{E} \left[ \sum_{h=0}^{\infty} \gamma^h r(s_h, a_h) \middle| s_0 = s, a_h \sim \pi(s_h), s_{h+1} \sim P(\cdot | s_h, a_h) \right]
\]

\[
V^\pi(s) = \mathbb{E}_{a \sim \pi(s)} \left[ r(s, a) + \gamma \mathbb{E}_{s' \sim P(\cdot | s, a)} V^\pi(s') \right]
\]

\[
Q^\pi(s, a) = \mathbb{E} \left[ \sum_{h=0}^{\infty} \gamma^h r(s_h, a_h) \middle| (s_0, a_0) = (s, a), a_h \sim \pi(s_h), s_{h+1} \sim P(\cdot | s_h, a_h) \right]
\]

\[
Q^\pi(s, a) = r(s, a) + \gamma \max_{a'} V^\pi(s')
\]
Bellman Equation:

\[
V^\pi(s) = \mathbb{E} \left[ \sum_{h=0}^{\infty} \gamma^h r(s_h, a_h) \mid s_0 = s, a_h \sim \pi(s_h), s_{h+1} \sim P(\cdot \mid s_h, a_h) \right]
\]

\[
V^\pi(s) = \mathbb{E}_{a \sim \pi(s)} \left[ r(s, a) + \gamma \mathbb{E}_{s' \sim P(\cdot \mid s, a)} V^\pi(s') \right]
\]

\[
Q^\pi(s, a) = \mathbb{E} \left[ \sum_{h=0}^{\infty} \gamma^h r(s_h, a_h) \mid (s_0, a_0) = (s, a), a_h \sim \pi(s_h), s_{h+1} \sim P(\cdot \mid s_h, a_h) \right]
\]

\[
Q^\pi(s, a) = r(s, a) + \gamma \mathbb{E}_{s' \sim P(\cdot \mid s, a)} V^\pi(s')
\]
Outline

1. Definition of infinite horizon discounted MDPs

2. Bellman Optimality

3. State-action distribution
Optimal Policy

For infinite horizon discounted MDP, there exists a deterministic stationary policy
\[ \pi^* : S \mapsto A, \text{ s.t., } V_{\pi^*}(s) \geq V_\pi(s), \forall s, \pi \]
[Puterman 94 chapter 6, also see theorem 1.4 in the RL monograph]
Optimal Policy

For infinite horizon discounted MDP, there exists a deterministic stationary policy
\[ \pi^* : S \mapsto A, \text{ s.t., } V^{\pi^*}(s) \geq V^{\pi}(s), \ \forall s, \pi \]
[Puterman 94 chapter 6, also see theorem 1.4 in the RL monograph]

We denote \( V^* := V^{\pi^*}, Q^* := Q^{\pi^*} \)
Optimal Policy

For infinite horizon discounted MDP, there exists a deterministic stationary policy

\[ \pi^*: S \mapsto A, \text{ s.t., } V^{\pi^*}(s) \geq V^\pi(s), \forall s, \pi \]

[Puterman 94 chapter 6, also see theorem 1.4 in the RL monograph]

We denote \( V^*: = V^{\pi^*}, Q^*: = Q^{\pi^*} \)

**Theorem 1: Bellman Optimality**

\[ V^*(s) = \max_a \left[ r(s, a) + \gamma \mathbb{E}_{s' \sim P(\cdot | s, a)} V^*(s') \right] \]
**Theorem 1: Bellman Optimality**

\[ V^*(s) = \max_a \left[ r(s, a) + \gamma \mathbb{E}_{s' \sim P(\cdot | s, a)} V^*(s') \right] \]
Theorem 1: Bellman Optimality

\[ V^*(s) = \max_a \left[ r(s, a) + \gamma \mathbb{E}_{s' \sim P(.|s,a)} V^*(s') \right] \]

Denote \( \hat{\pi}(s) := \arg \max_a Q^*(s, a) \), we will prove \( V^{\hat{\pi}}(s) = V^*(s), \forall s \)
Proof of Bellman Optimality

**Theorem 1: Bellman Optimality**

\[ V^*(s) = \max_a \left[ r(s, a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s, a)} V^*(s') \right] \]

Denote \( \hat{\pi}(s) := \arg \max_a Q^*(s, a) \), we will prove \( V^*(s) = V^*(s), \forall s \)

\[ V^*(s) = r(s, \pi^*(s)) + \gamma \mathbb{E}_{s' \sim P(s, \pi^*(s))} V^*(s') \]
Theorem 1: Bellman Optimality

\[ V^*(s) = \max_a \left[ r(s, a) + \gamma \mathbb{E}_{s' \sim P(\cdot | s, a)} V^*(s') \right] \]

Denote \( \hat{\pi}(s) := \arg \max_a Q^*(s, a) \), we will prove \( V^\hat{\pi}(s) = V^*(s), \forall s \)

\[
V^*(s) = r(s, \pi^*(s)) + \gamma \mathbb{E}_{s' \sim P(s, \pi^*(s))} V^*(s') \\
\leq \max_a \left[ r(s, a) + \gamma \mathbb{E}_{s' \sim P(s, a)} V^*(s') \right] = r(s, \hat{\pi}(s)) + \gamma \mathbb{E}_{s' \sim P(s, \hat{\pi}(s))} V^*(s')
\]
Proof of Bellman Optimality

**Theorem 1: Bellman Optimality**

\[ V^*(s) = \max_a \left[ r(s, a) + \gamma \mathbb{E}_{s' \sim P(\cdot | s, a)} V^*(s') \right] \]

Denote \( \hat{\pi}(s) := \arg \max_a Q^*(s, a) \), we will prove \( V^*(s) = V^*(s), \forall s \)

\[
V^*(s) = r(s, \pi^*(s)) + \gamma \mathbb{E}_{s' \sim P(s, \pi^*(s))} V^*(s') \\
\leq \max_a \left[ r(s, a) + \gamma \mathbb{E}_{s' \sim P(s, a)} V^*(s') \right] = r(s, \hat{\pi}(s)) + \gamma \mathbb{E}_{s' \sim P(s, \hat{\pi}(s))} V^*(s') \\
= r(s, \hat{\pi}(s)) + \gamma \mathbb{E}_{s' \sim P(s, \hat{\pi}(s))} \left[ r(s', \pi^*(s')) + \gamma \mathbb{E}_{s'' \sim P(s', \pi^*(s'))} V^*(s'') \right]
\]
Proof of Bellman Optimality

**Theorem 1: Bellman Optimality**

\[ V^*(s) = \max_a \left[ r(s, a) + \gamma \mathbb{E}_{s' \sim P(\cdot | s, a)} V^*(s') \right] \]

Denote \( \hat{\pi}(s) := \arg \max_a Q^*(s, a) \), we will prove \( V^*(s) = V^*(s), \forall s \)

\[
V^*(s) = r(s, \pi^*(s)) + \gamma \mathbb{E}_{s' \sim P(s, \pi^*(s))} V^*(s') \\
\leq \max_a \left[ r(s, a) + \gamma \mathbb{E}_{s' \sim P(s, a)} V^*(s') \right] = r(s, \hat{\pi}(s)) + \gamma \mathbb{E}_{s' \sim P(s, \hat{\pi}(s))} V^*(s') \\
= r(s, \hat{\pi}(s)) + \gamma \mathbb{E}_{s' \sim P(s, \hat{\pi}(s))} \left[ r(s', \pi^*(s')) + \gamma \mathbb{E}_{s'' \sim P(s', \pi^*(s'))} V^*(s'') \right] \\
\leq r(s, \hat{\pi}(s)) + \gamma \mathbb{E}_{s' \sim P(s, \hat{\pi}(s))} \left[ r(s', \hat{\pi}(s')) + \gamma \mathbb{E}_{s'' \sim P(s', \hat{\pi}(s'))} V^*(s'') \right]
\]
Proof of Bellman Optimality

**Theorem 1: Bellman Optimality**

\[ V^*(s) = \max_a \left[ r(s, a) + \gamma \mathbb{E}_{s' \sim P(s'|s, a)} V^*(s') \right] \]

Denote \( \hat{\pi}(s) := \arg \max_a Q^*(s, a) \), we will prove \( V^*(s) \)

\[
V^*(s) = r(s, \pi^*(s)) + \gamma \mathbb{E}_{s' \sim P(s,\pi^*(s))} V^*(s') \\
\leq \max_a \left[ r(s, a) + \gamma \mathbb{E}_{s' \sim P(s,a)} V^*(s') \right] = r(s, \hat{\pi}(s)) + \gamma \mathbb{E}_{s' \sim P(s,\hat{\pi}(s))} V^*(s') \\
= r(s, \hat{\pi}(s)) + \gamma \mathbb{E}_{s' \sim P(s,\hat{\pi}(s))} \left[ r(s', \pi^*(s')) + \gamma \mathbb{E}_{s'' \sim P(s',\pi^*(s'))} V^*(s'') \right] \\
\leq r(s, \hat{\pi}(s)) + \gamma \mathbb{E}_{s' \sim P(s,\hat{\pi}(s))} \left[ r(s', \hat{\pi}(s')) + \gamma \mathbb{E}_{s'' \sim P(s',\hat{\pi}(s'))} V^*(s'') \right] \\
\leq r(s, \hat{\pi}(s)) + \gamma \mathbb{E}_{s' \sim P(s,\hat{\pi}(s))} \left[ r(s', \hat{\pi}(s')) + \gamma \mathbb{E}_{s'' \sim P(s',\hat{\pi}(s'))} V^*(s'') \right]
\]
Proof of Bellman Optimality

**Theorem 1: Bellman Optimality**

\[
V^*(s) = \max_a \left[ r(s, a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s, a)} V^*(s') \right]
\]

Denote \( \hat{\pi}(s) := \arg \max_a Q^*(s, a) \), we will prove \( V^\pi(s) = V^*(s) \), \( \forall s \)

\[
V^*(s) = r(s, \pi^*(s)) + \gamma \mathbb{E}_{s' \sim P(s, \pi^*(s))} V^*(s') \\
\leq \max_a \left[ r(s, a) + \gamma \mathbb{E}_{s' \sim P(s, a)} V^*(s') \right] = r(s, \hat{\pi}(s)) + \gamma \mathbb{E}_{s' \sim P(s, \hat{\pi}(s))} V^*(s') \\
= r(s, \hat{\pi}(s)) + \gamma \mathbb{E}_{s' \sim P(s, \hat{\pi}(s))} \left[ r(s', \pi^*(s')) + \gamma \mathbb{E}_{s'' \sim P(s', \pi^*(s'))} V^*(s'') \right] \\
\leq r(s, \hat{\pi}(s)) + \gamma \mathbb{E}_{s' \sim P(s, \hat{\pi}(s))} \left[ r(s', \hat{\pi}(s')) + \gamma \mathbb{E}_{s'' \sim P(s', \hat{\pi}(s'))} V^*(s'') \right] \\
\leq r(s, \hat{\pi}(s)) + \gamma \mathbb{E}_{s' \sim P(s, \hat{\pi}(s))} \left[ r(s', \hat{\pi}(s')) + \gamma \mathbb{E}_{s'' \sim P(s', \hat{\pi}(s'))} V^*(s'') \right] \\
\leq \mathbb{E} \left[ r(s, \hat{\pi}(s)) + \gamma r(s', \hat{\pi}(s')) + \ldots \right] = V^\hat{\pi}(s)
\]