Mathematics of Reinforcement Learning

Alexey Naumov

HDI Lab,
HSE University

August 2-23, 2020
Summary

1. Introduction

2. Formalizing RL Problems

3. TD learning with function approximation

4. Q-learning
Consider the following setting.

- **Environment**: described at moment $t \in \mathbb{Z}$ by $s_t \in S$, *state space*;
- **Agent** performs actions $a_t \in A$ available in *action space* by exploiting $s_t$ with decision rule (*policy*). It changes the environment to $s_{t+1}$ and agent receives reward $r_{t+1} \in \mathbb{R}$. Goal of the agent is to maximize his expected reward with respect to his policy.

Unfortunately, there is a big problem:

**Problem**

The dynamics of the system is not known in advance.
Some Applications

- Robot tasks (grab the thing, walk a step...) Zhu et al. (2020).
- Online advertisement (RTB-auctions) Liu et al. (2019).
- Thermal control of crystal growth Beintema et al. (2020).
- AI parts for computer games (especially, micro-level) Shao et al. (2018); Barriga et al. (2019).
- There are benchmarks for testing, the most popular is OpenAI Gym https://gym.openai.com
OpenAI Gym: classical control

Classic control
Control theory problems from the classic RL literature.

Acrobot-v1
Swing up a two-link robot.

CartPole-v1
Balance a pole on a cart.

MountainCar-v0
Drive up a big hill.

MountainCarContinuous-v0
Drive up a big hill with continuous control.

Pendulum-v0
Swing up a pendulum.
OpenAI Gym: ATARI

Atari
Reach high scores in Atari 2600 games.
Challenges of RL

1. The dynamics may be completely unknown.
2. The data collection may be very expensive.
3. Need for data in RL considerably exceeds the one in "usual" machine learning settings.
4. Hard to achieve stability and robustness of learned policies.
5. Statistical analysis is extreme even in simpler settings – the data are usually not i.i.d.
References


More general theory: Michel Benaïm, Dynamics of stochastic approximation algorithms, Springer (Benaïm (1999))
Markov Decision Process (MDP)

- \( S \) be state space. By \((S_t)_{t \geq 0}\) we denote a sequence of random states.
- \( \mathcal{A} \) action space. Let \((A_t)_{t \geq 0}\) be a sequence of random actions.
- A random policy \( \pi \) is the sequence of distributions on \( \mathcal{A} \)
  \[
  \pi_t(a|s) = \mathbb{P}(A_t = a|S_t = s)
  \]
- Markov kernel \( P_t(s'|s, a) := \mathbb{P}(S_t = s'|S_{t-1} = s, A_{t-1} = a) \).
- (Deterministic) reward \( R : S \times \mathcal{A} \to \mathbb{R} \)
- At step \( t \) in the state \( S_t = s \) the agent performs an action \( A_t \sim \pi_t(\cdot|s) \) (let \( A_t = a \)), obtains a reward \( R_{t+1} = R(S_t, A_t) \) and transits to \( S_{t+1} \sim P_t(\cdot|s, a) \) (there are different formulations, e.g. \( R_{t+1} \) could be function of \( S_{t+1} \) as well)

Markov Decision Process

Let \( \gamma \in (0, 1] \). Tuple \((S, \mathcal{A}, P, R, \gamma)\) is called Markov Decision Process.
Note that

$$P^\pi_t(s'|s) = \sum_{a \in A} P_t(s'|s, a) \pi_{t-1}(a|s)$$

is the Markov kernel (associated with $\pi$) and $(S_t)_{t \geq 0}$ – MC

Path distribution: for some $T > t$

$$\mathbb{P}(A_t = a_t, S_{t+1} = s_{t+1}, \ldots, S_T = s_T, A_T = a_T | S_t = s_t)$$

$$= \pi_t(a_t|s_t) \prod_{k=t+1}^{T} P_k(s_k|s_{k-1}, a_{k-1}) \pi_k(a_k|s_k)$$
Let $\Pi$ be a set of policies.

Fix some policy $\pi \in \Pi$. The $t$-value function $V_{\pi,t}(s)$ for a state $s$ at $t$ is

$$V_{\pi,t}(s) := \mathbb{E}\left[ \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s \right]$$

Further on, assume time-homogeneous kernels, then for any $k > 1$

$$V_{\pi,t}(s) = V_{\pi,t+k}(s) =: V_{\pi}(s)$$
Recursive relation: Value function

Note that

\[ V_{\pi, t}(S_t) = E\left[ \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t \right] = E \left[ R_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k R_{t+k+2} | S_t \right] \]

\[ = E \left[ R_{t+1} | S_t \right] + \gamma E \left[ \sum_{k=0}^{\infty} \gamma^k R_{t+k+2} | S_{t+1} \right] \]

\[ = E \left[ R_{t+1} | S_t \right] + \gamma E \left[ \sum_{k=0}^{\infty} \gamma^k R_{t+k+2} | S_{t+1} \right] \]

\[ = E \left[ R_{t+1} | S_t \right] + \gamma E \left[ V_{\pi, t+1}(S_{t+1}) | S_t \right] \]

Homogeneous case (for simplicity)

\[ V_{\pi}(s) = E[R(S_0, A_0) | S_0 = s] + \gamma P^\pi V_{\pi}(s) \]
### Action-Value Functions

Let $\Pi$ be a set of policies.

**Action-Value Functions**

Fix some policy $\pi \in \Pi$. The $t$-value function $Q_{\pi,t}(s)$ for a state $s$ at $t$ is

$$Q_{\pi,t}(s, a) := E \left[ \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s, A_t = a \right]$$

**Homogeneous kernels:** for any $k > 1$

$$Q_{\pi,t}(s, a) = Q_{\pi,t+k}(s, a) =: Q_\pi(s, a)$$

**Recursive relation**

$$Q_\pi(s, a) = R(s, a) + \gamma E[Q_\pi(S_1, A_1) \mid S_0 = s, A_0 = a]$$

**Exercise:** write down relation between $V_\pi$ and $Q_\pi$
Finite case

- **Value function**

\[
V_\pi(s) = \sum_{a \in A} R(s, a)\pi(a|s) + \gamma \sum_{s' \in S, a \in A} V_\pi(s')\pi(a|s)P(s'|s, a) \\
= \sum_{a \in A} \pi(a|s)\left(R(s, a) + \gamma \sum_{s' \in S} V_\pi(s')P(s'|s, a)\right)
\]

- **Action-Value function**

\[
Q_\pi(s, a) = R(s, a) + \gamma \sum_{s' \in S} V_\pi(s')P(s'|s, a) \\
= R(s, a) + \gamma \sum_{s' \in S, a' \in A} Q_\pi(s', a')P(s'|s, a)\pi(a'|s')
\]
Idea of policy improvement

We say that $\pi' \succeq \pi$ if $V_{\pi'}(s) \geq V_{\pi}(s)$ for any $s \in S$

Optimal policy

$\pi^*$ is optimal policy if for any $\pi \in \Pi$

$$V_{\pi^*}(s) \geq V_{\pi}(s) \text{ for all } s \in S$$
Idea of policy improvement

▶ Fix $\pi$ and recall:

$$V_\pi(s) = \sum_{a \in A} \pi(a|s) (R(s, a) + \gamma \sum_{s' \in S} V_\pi(s') P(s'|s, a))$$

▶ Suppose that there is some action $a'$, such that:

$$R(s, a') + \gamma \sum_{s' \in S} V_\pi(s') P(s'|s, a') > V_\pi(s)$$

▶ Then take

$$\pi'(a|s) = \begin{cases} 1, & \text{if } a = a', \\ 0, & \text{otherwise} \end{cases}$$

▶ For current $s$: $V_{\pi'}(s) \geq V_\pi(s)$
More generally, we can change the policy \( \pi \) to a new policy \( \pi' \), which is greedy with respect to the computed values \( V_\pi \)

\[
\pi'(s) := \arg \max_{a \in \mathcal{A}} \left( R(s, a) + \gamma \sum_{s' \in S} V_\pi(s') P(s'|s, a) \right)
\]

\[
= \arg \max_{a \in \mathcal{A}} Q_\pi(s, a)
\]

Then for current \( s \): \( V_{\pi'}(s) \geq V_\pi(s) \)
Optimal state value

<table>
<thead>
<tr>
<th>Optimal Value Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ V^*(s) := \max_{\pi \in \Pi} V_\pi(s) ]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Optimal Action-Value Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ Q^*(s, a) := \max_{\pi \in \Pi} Q_\pi(s, a) ]</td>
</tr>
</tbody>
</table>

- There exists a unique optimal value function (shown by Bellman, 1957)
- \[ V_{\pi^*}(s) = V^*(s), Q_{\pi^*}(s, a) = Q^*(s, a) \]
Bellman Equations

▶ Take

$$\pi^*(a|s) = \begin{cases} 1, & \text{if } a = \arg \max_{a \in A} Q^*(s, a), \\ 0, & \text{otherwise} \end{cases}$$

▶ Then

$$V^*(s) = \max_a Q^*(s, a) = \max_a \mathbb{E}[R_{t+1} + \gamma V^*(S_{t+1})|S_t = s, A_t = a]$$

Optimality Equations

$$V^*(s_t) = \max_{a_t \in A} \{ R_{t+1} + \gamma \mathbb{E}[V^*(S_{t+1})|S_t = s_t, A_t = a_t] \}.$$ 

$$Q^*(s_t, a_t) = R(s_t, a_t) + \gamma \mathbb{E}[\max_{a \in A} Q^*(S_{t+1}, a)|S_t = s_t, A_t = a_t]$$
Bellman Iterations

**Theorem**

Assume $V : S \rightarrow \mathbb{R}$ be some function from Banach space $\mathcal{B}$ with norm $\| \cdot \|$. Then operators $T^\pi, T^* : \mathcal{B} \rightarrow \mathcal{B}$ defined as

$$
T^\pi : V \mapsto E[R(s, A_t)|S_t = s] + \gamma E[V(S_{t+1})|S_t = s] =: T^\pi[V](s),
$$

$$
T^* : V \mapsto \max_{a \in A} \{R(s, a) + \gamma E[V(S_{t+1})|S_t = s, A_t = a]\} =: T^*[V](s)
$$

are contractions

$$
\| T^\pi(V) - T^\pi(V') \| \leq \gamma \| V - V' \|
$$

Moreover, their fixed points are $V_\pi$ and $V^*$ for an optimal policy $\pi^*$. 
Two Problems for MDP

- Policy Evaluation: compute $V_{\pi}$ given fixed policy $\pi \in \Pi$.
- Policy improvement: compute or approximate some optimal policy $\pi^*$, solve control problem.

Both could be solved with Bellman equations using fixed point iteration...

**BUT**

Even if transition model $P(\cdot | s, a)$ is known, the expectation in right part is often intractable!
Policy iteration: The Simplest Case

- Consider tabular case: $|S|, |A| < \infty$ and known transition matrix $P^\pi$ for Markov chain $S_t$. The value function can be represented as vector $V \in \mathbb{R}^{|S|}$.

- Policy evaluation: for given $\pi$

$$V^{(k+1)}_\pi(s) = E[R(S_0, A_0)|S_0 = s] + \gamma P^\pi V^{(k)}_\pi(s)$$

Fact: $V^{(k)}_\pi \to V_\pi$ as $k \to \infty$ in $\| \cdot \|_\infty$. Prove it.

- Policy improvement:

$$\pi(a|s) := \delta_{a^\pi(s)}$$

where

$$a^\pi(s) := \arg \max_{a \in A} \{ R(s, a) + \gamma E[V^{(k)}_\pi(S_1)|S_0 = s, A_0 = a] \}$$

for some large $k$.

Problems

- Computational problems: Even in finite case $|S|$ may be extremely large (see chess,...).

- Algorithmic problems: In infinite case the iterative procedure is intractable or cannot be correctly built.
Policy evaluation

Recall the Bellman expectation equation for value function $V_\pi$:

$$V_\pi(s) = E[R(S_0, A_0) | S_0 = s] + \gamma P^\pi V_\pi(s)$$

How can we estimate this quantity?

1. **Monte-Carlo.** Run a series of independent simulations to compute $V_\pi(s) = E [\sum_{t=0}^{\infty} \gamma^t R_{t+1} | S_0 = s]$.

2. **Stochastic Approximation.** Use Robbins-Monro procedure and Bellman equation to obtain $TD(0)$. Much more efficient and preliminary estimates are available even at the early stage.
TD(0) with function approximation

- We consider approximation of the value function $V : S \to \mathbb{R}$ using a parameterized function $V : S \times \mathbb{R}^d \to \mathbb{R}$, \((s, \theta) \mapsto V(x, \theta)\) where $\theta$ is a vector of parameters.

- Minimize the root-mean-squared error (RMSE) over some distribution $d$ over the inputs

\[
\text{RMSR}(\theta) = \sqrt{\sum_{s \in S} d(s)(V_\pi(s) - V(s, \theta))^2},
\]

- Suppose that we observe a sequence of states $\{S_t\}_{t \geq 0}$ (generated according to $P^\pi$), and that at time $t$, the vector of parameters is denoted as $\theta_t$.

- Use gradient descent

\[
\theta_{t+1} = \theta_t - \frac{\alpha}{2} \nabla (V_\pi(S_t) - V(S_t, \theta_t))^2
= \theta_t + \alpha (V_\pi(S_t) - V(S_t, \theta_t)) \nabla V(S_t, \theta_t)
\]
TD(0) with function approximation

- Replace $V_\pi(S_t)$ by its estimator
- Backward view:

$$\theta_{t+1} = \theta_t + \alpha_{t+1}\left\{ \overline{R}_{t+1} + \gamma V(S_{t+1}, \theta_t) - V(S_t, \theta_t) \right\}\nabla V(S_t, \theta_t), \quad (1)$$

here $\overline{R}_{t+1} = \overline{R}(S_t) = E[R(S_t, A_t)|S_t]$

- Linear function approximation:

$$V(s, \theta) = \theta^\top \psi(s)$$

where $\psi(s) = [\psi^1(s), \ldots, \psi^d(s)]^\top$. The vector $\psi(s)$ is referred to as the feature vector associated to the state $s \in S$.

- The gradient of the approximate value function in such case is

$$\nabla V(s, \theta) = \psi(s)$$
Define for $t \geq 0$, $Z_t = [S_{t-1}, S_t]^\top$. We may rewrite (1) as

$$\theta_{t+1} = \{I - \alpha_{t+1} \tilde{A}(Z_{t+1})\} \theta_t + \alpha_{t+1} \tilde{b}(Z_{t+1}),$$

where $\tilde{A}(z)$ is a $d \times d$ matrix given for $z = [s, s']^\top \in S^2$ by

$$\tilde{A}(z) = \psi(s)\{\psi(s) - \gamma\psi(s')\}^\top,$$

and $\tilde{b}(z)$ is a $d \times 1$ vector given by

$$\tilde{b}(z) = \tilde{R}(s)\psi(s).$$
Note that \( \{Z_t\}_{t=0}^{\infty} \) is a Markov chain on the state-space

\[
Z = \{ z = (x_0, x_1) \in S^2, Q(s_0, s_1) > 0 \}.
\]  \hfill (4)

The transition matrix of this Markov chain is given, for any \((s_0, s_1), (s'_0, s'_1) \in Z\), by

\[
P(s_0, s_1; s'_0, s'_1) = \delta_{s_1, s'_0} Q(s_1, s'_1),
\]

where \( \delta_{u,v} \) is the Kronecker symbol.

It is easily seen that this \( P \) has a unique invariant distribution given by

\[
\bar{\pi}(s_0, s_1) = \bar{\pi}_0(s_0) P^\pi(s_0, s_1),
\]

where \( \pi_0 \) is stationary distribution of \( P^\pi \) (we assume that \( \bar{\pi}_0 \) exists).
We would like to solve

\[ F(x) = 0, \quad F : \mathbb{R}^m \rightarrow \mathbb{R}^n, \]

for some reason we cannot compute \( F(x) \) at any desirable point and have solely some unbiased noisy estimate \( F(x, \xi) \) such that \( \mathbb{E}[F(x, \xi)] = F(x) \).

Robbins and Monro (1951):

1. Set some \( X_0 \in \mathbb{R}^m \).
2. Recompute as long as you need with some decreasing (or fixed) stepsize \( \alpha_k \)
   \[ X_{k+1} = X_k + \alpha_k F(X_k, \xi_{k+1}). \]

The procedure converges under mild conditions!

Compare with the standard ‘Euler scheme’ for numerically approximating a trajectory of the o.d.e. \( \dot{x}(t) = F(x(t)) \)

\[ x_{t+1} = x_t + \alpha F(x_t) \]
Robbins-Monro procedure, linear case

- **Aim:** find $\theta^* \in \mathbb{R}^d : A\theta^* = b$ from noisy observations
  \[
  \{(\bar{A}(Z_n), \bar{b}(Z_n))\}_{n \in \mathbb{N}}, \quad \text{where } \{Z_n\} \text{ is a sequence of random variables}
  \]
  taking values in a general state-space $Z$, and $\bar{A} : Z \to \mathbb{R}^{d \times d}$, $\bar{b} : Z \to \mathbb{R}^d$

- **Linear SA:**
  \[
  \theta_{n+1} = \theta_n - \alpha_{n+1}\{(\bar{A}(Z_{n+1})\theta_n - \bar{b}(Z_{n+1}))
  \]
  where $\{\alpha_n\}$ is a non-negative stepsize sequence.

- $\{Z_n\}_{n \in \mathbb{N}}$ is an ergodic Markov chain with kernel $\bar{P}$ and unique stationary distribution $\bar{\pi}$ satisfying $\lim_{n \to \infty} E[\bar{A}(Z_n)] = A$ and $\lim_{n \to \infty} E[\bar{b}(Z_n)] = b$. 
Assumptions (A)

- The Markov kernel $\bar{P}$ is uniformly ergodic, i.e., there exist $\rho \in [0, 1)$ and $C_\rho < \infty$ such that $\sup_{z \in \mathbb{Z}} \| \bar{P}^n(z, \cdot) - \bar{\pi} \|_{TV} \leq C_\rho \rho^n$ for any $n \in \mathbb{N}$.

- There exists a constant $\bar{C}_b > 0$ such that $\sup_{z \in \mathbb{Z}} \| \bar{b}(z) \| \leq \bar{C}_b$.

- There exists a constant $\bar{C}_A > 0$ such that $\sup_{z \in \mathbb{Z}} \| \bar{A}(z) \| \leq \bar{C}_A$.

- The square matrix $-A = -E_{\bar{\pi}}[\bar{A}(Z_0)]$ is Hurwitz ($\text{Re}(\lambda(-A)) < 0$). There exists a positive definite matrix $B$ satisfying the Lyapunov equation

$$A^\top B + BA = I.$$ 

If $A^\top = A$, take $B = A^{-1}/2$. Any positive definite matrix $A$ is Hurwitz.

- Fixed step size or

$$\sum_{k=1}^{\infty} \alpha_k = \infty, \quad \sum_{k=1}^{\infty} \alpha_k^2 < \infty.$$
**Theorem**

*Under assumptions (A)*

\[ E[\|\theta_n - \theta^*\|^2] \leq C_1 e^{-ca \sum_{k=1}^n \alpha_k} E[\|\theta_0 - \theta^*\|^2] + C_2 \alpha_n, \]

where \( c, C_1, C_2 > 0 \) are some constants and

\[ a := \|B\|^{-1}. \]

- Holds for more general class of MC (e.g. \( V\)-ergodic)
- See e.g. (E. Moulines, A. Naumov, S. Samsonov, H.T. Wai, On linear stochastic approximation with Markovian noise: Fine-grained analysis finite time error analysis, submitted)
L₂ convergence of LSA

(a)

(c)

(d)

MSE error, TD(0)
TD(0): Check that $-A$ is Hurwitz

- We check that
  \[ \text{Re} \lambda(A) = \text{Re} \lambda(E[\psi(S)\{\psi(S) - \gamma\psi(S')\}^\top]) > 0 \]

- Fix $x$ and use Cauchy-Schwarz inequality
  \[ |E[x^\top \psi(S)\psi(S')^\top x]| \leq \sqrt{E[x^\top \psi(S)\psi(S)^\top x]E[x^\top \psi(S')\psi(S')^\top x]} \]
  \[ = E[x^\top \psi(S)\psi(S)^\top x] \]

- Hence,
  \[ x^\top E[\psi(S)\{\psi(S) - \gamma\psi(S')\}^\top]x \geq (1 - \gamma)E[x^\top \psi(S)\psi(S)^\top x] > 0 \]

- Let $\lambda = \mu + i\nu$ - eigenvalue of $A$. Then
  \[ (A - \lambda I)(x + iy) = (A - \mu I)x + \nu y + i(-\nu x + (A - \mu I)y) = 0, \]
  \[ x^\top (A - \mu I)x = -\nu x^\top y, \ y^\top (A - \mu I)y = \nu y^\top x, \]
  \[ x^\top (A - \mu I)x + y^\top (A - \mu I)y = 0, \]
  \[ \mu = \frac{x^\top Ax + y^\top Ay}{x^\top x + y^\top y} > 0 \]
Proof of LSA + Project

- Recall

\[ \theta_{n+1} = (I - \alpha_{n+1} \bar{A}(Z_{n+1}))\theta_n + \alpha_{n+1} \bar{b}(Z_{n+1}) \]

- Recursion for the error \( \tilde{\theta}_n = \theta_n - \theta^* \) holds:

\[ \tilde{\theta}_{n+1} = \{I - \alpha_{n+1} \bar{A}(Z_{n+1})\} \tilde{\theta}_n + \alpha_{n+1} \bar{\epsilon}(Z_{n+1}), \quad \bar{\epsilon}(z) := \bar{b}(z) - \bar{A}(z)\theta^*. \]

where we have set \( \tilde{A}(z) := \bar{A}(z) - A, \tilde{b}(z) := \bar{b}(z) - b. \)

- \( \tilde{\theta}_n = \tilde{\theta}^{(\text{tr})}_n + \tilde{\theta}^{(\text{fl})}_n \), where \( \tilde{\theta}^{(\text{tr})}_n \) is the transient term and \( \tilde{\theta}^{(\text{fl})}_n \) is the fluctuation term, defined as

\[
\begin{align*}
\tilde{\theta}^{(\text{tr})}_{n+1} &= \{I - \alpha_{n+1} \bar{A}(Z_{n+1})\} \tilde{\theta}^{(\text{tr})}_n, & \tilde{\theta}^{(\text{tr})}_0 &= \tilde{\theta}_0, \\
\tilde{\theta}^{(\text{fl})}_{n+1} &= \{I - \alpha_{n+1} \bar{A}(Z_{n+1})\} \tilde{\theta}^{(\text{fl})}_n + \alpha_{n+1} \bar{\epsilon}(Z_{n+1}), & \tilde{\theta}^{(\text{fl})}_0 &= 0.
\end{align*}
\]

- Replace the random matrix \( \bar{A}(Z_{n+1}) \) by \( A \), \( \tilde{\theta}^{(\text{fl})}_{n+1} = J^{(0)}_{n+1} + H^{(0)}_{n+1} \)

\[
\begin{align*}
J^{(0)}_{n+1} &= (I - \alpha_{n+1} A) J^{(0)}_n + \alpha_{n+1} \bar{\epsilon}(Z_{n+1}), & J^{(0)}_0 &= 0, \\
H^{(0)}_{n+1} &= (I - \alpha_{n+1} \bar{A}(Z_{n+1})) H^{(0)}_n - \alpha_{n+1} \tilde{A}(Z_{n+1}) J^{(0)}_n, & H^{(0)}_0 &= 0.
\end{align*}
\]
Polyak-Ruppert averaging procedure

- Polyak-Ruppert averaged sequence, \((\bar{\theta}_n)_{n \geq 1}\),

\[
\bar{\theta}_n = \frac{1}{n} \sum_{k=0}^{n-1} \theta_k
\]

- It is known that for suitably decaying step sizes, a central limit theorem (CLT) can be established for the averaged iterates.

- Moreover, Polyak-Ruppert averaging can achieve an optimal covariance, in the sense of local asymptotic minimaxity.

- Project: Study

\[
E[\|\bar{\theta}_n - \theta^*\|^2]
\]
Gradient TD Principle

- Recall TD(0)

\[ \theta_{t+1} = \theta_t + \alpha_{t+1} \{ \overline{R}_{t+1} + \gamma \theta_t^\top \psi(S_{t+1}) - \theta_t^\top \psi(S_t) \} \nabla V(S_t, \theta_t) \]

- TD-error: \( D_t(\theta_t) = \overline{R}_{t+1} + \gamma \theta_t^\top \psi_t' - \theta_t^\top \psi_t \), where \( \psi_t = \psi(S_t), \psi_t' = \psi(S_{t+1}) \)

- The linear TD solution \( \theta^* \) shall satisfy

\[ 0 = \mathbb{E}[\psi \cdot D(\theta^*)] = \lim_{t \to \infty} \mathbb{E}[\psi(S_t) \cdot D_t(\theta^*)] = -A \theta^* + b \]

where we recall

\[ A := \lim_{t \to \infty} \mathbb{E}[\psi(S_t)\{\psi(S_t) - \gamma \psi(S_{t+1})\}^\top] = \mathbb{E}[\psi\{\psi - \gamma \psi'\}^\top] \]

\[ b := \lim_{t \to \infty} \mathbb{E}[\overline{R}_{t+1} \psi(S_t)] = \mathbb{E}[\overline{R} \psi]. \]

- In GTD0, we consider the objective function given as the norm of expected TD update:

\[ J(\theta) = (1/2) \| \mathbb{E}[\psi \cdot D(\theta)] \|^2 = (1/2) \| b - A \theta \|^2 \]
The gradient of the objective function is

\[ \nabla J(\theta) = A^T (A\theta - b) = -E[\{\phi - \alpha \psi'\}\psi^T]E[\psi \cdot D(\theta)] \]

A naive gradient estimator as \( \{(\psi_t - \gamma \psi_{t+1})\psi_t^T\}\{\psi_t \cdot D_t(\theta_t)\} \) does not work as it gives a biased estimate of \( J(\theta) \).

GTD0

Linear two time scale stochastic approximation:

\[
\begin{align*}
\theta_{t+1} &= \theta_t + \beta_t \{\psi_t - \gamma \psi_{t+1}\}\psi_t^T \psi_t \\
\psi_t &= \psi_t - \theta_t^T \psi_t - w_t.
\end{align*}
\]

We set \( \beta_k / \gamma_k \to 0 \) and \( w_t \) is ‘almost’ stationary w.r.t. \( \theta_t \).

See Kaledin et al. (2020)
Experiments: Garnet Problem with Markovian Samples

Key parameters:

1. A problem from the Garnet family $n_S = 50$, $n_A = 10$, $b = 2$;
2. Stepsizes $\beta_k = c^\beta / (k + k_0^\beta)$, $\gamma_k = c^\gamma / (k + k_0^\gamma)^{2/3}$

Figure: Deviations from stationary point $(\theta^*, \omega^*)$
Let \((S, A, P, R, \gamma)\) be MDP and \(\Pi\) be some set of policies.

**Control Problem**

Maximize \(E \left[ \sum_{t=0}^{\infty} \gamma^t R_t \right]\) with respect to \(\pi \in \Pi\). The whole dynamics of \(S_t, A_t\) depends on the choice of \(\pi\)!

- If \(\pi^*_\) is an optimal policy (i.e. for any \(\pi \in \Pi\) \(V^*_*(s) \geq V_{\pi}(s)\) in all states \(s \in S\)) then it also solves this (prove it!).
- There is **no fixed** policy: you have to choose the best of the available ones.
- ...despite this, could policy evaluation help?
Policy Iteration

1. Start with some deterministic policy $\pi_0 \in \Pi$.
2. As long as it is needed...
3. Run policy evaluation, obtain approximate $V$-function $\tilde{V}_{\pi_k}$.
4. Run policy improvement:

$$\pi_{k+1}(s) = \arg\max_{a \in A} \left\{ R(s, a) + \gamma \mathbb{E} \left[ \tilde{V}_{\pi_k}(S_1) | S_0 = s, A_0 = a \right] \right\}$$

Problem

The algorithm needs transition model!

So we have to handle that expectation with Monte Carlo in another way.
SARSA-Algorithm

Recall the Bellman expectation equation for $Q$-function:

$$Q_\pi(s_t, a_t) = r_{t+1} + \gamma \mathbb{E} \left[ Q_\pi(S_{t+1}, A_{t+1}) | S_t = s_t, A_t = a_t \right].$$

Idea: if known, the best immediate action is $a_t = \arg\max_{a \in A} Q_\pi(s_t, a)$. 

Greedy Policy (Recap)

Let $Q : S \times A \to \mathbb{R}$, policy $\pi_Q(a|s) = \delta_{\arg\max_{a \in A} Q(s, a)}$ is called greedy policy generated by $Q$. Shorter notation is $\pi_Q : S \to A$. 

Now use Robbins-Monro method!

**SARSA-algorithm**

1. Start with some initial estimate $Q_0 : S \times A \to \mathbb{R}$ and let $\pi = \pi_{Q_0}$.
2. For $k = 1, 2, ...$
3. 3.1 **Sample** trajectory $s_0, a_0, r_1, s_1, a_1, ..., r_T, s_T$ with $a_t = \pi(s_t)$.
3. 3.2 **Update Q.** For all $t \in 0, ..., T - 1$
   $$Q_k(s_t, a_t) = Q_{k-1}(s_t, a_t) + \alpha_k (r_{t+1} + \gamma Q_{k-1}(s_{t+1}, a_{t+1}) - Q_{k-1}(s_t, a_t)).$$
3. 3.3 **Update Policy.** $\pi = \pi_{Q_k}$.

This is an **on-policy** algorithm: we update $\pi_{Q_k}$ with samples from $\pi_{Q_k}$. 

Expected SARSA

Instead of estimate

\[ r_{t+1} + \gamma Q_{k-1}(s_{t+1}, a_{t+1}) \]

for the right part of the Bellman equation, use another. Exploit another policy \( \pi_b \)

\[ r_{t+1} + \gamma E[Q_{k-1}(s_{t+1}, A)] = r_{t+1} + \gamma \sum_{a \in A} \pi_b(a|s_t) Q_{k-1}(s_{t+1}, a). \]

where \( A \sim \pi_b(\cdot|s) \). So we get off-policy algorithm: we use another (fixed) policy \( \pi_b \) to update the estimate of greedy \( \pi Q_k \).
Q-Learning

We might recall Bellman’s optimality equation for $Q^*$-function:

$$Q^*(s_t, a_t) = R(s_t, a_t) + \gamma \mathbb{E}\left[\max_{a \in A} Q^*(S_{t+1}, a) | S_t = s_t, A_t = a_t\right]$$

and apply Robbins-Monro algorithm to this equation in $Q^*$.

**Q-Learning**

1. Start with some initial estimate $Q_0 : S \times A \rightarrow \mathbb{R}$ and let $\pi = \pi_{Q_0}$.
2. For $k = 1, 2, ...$
3. 3.1 Sample trajectory $s_0, a_0, r_1, s_1, a_1, ..., r_T, s_T$ with $a_t = \pi(s_t)$.
   3.2 Update Q. For all $t \in 0, .., T - 1$

   $$Q_k(s_t, a_t) = Q_{k-1}(s_t, a_t) + \alpha_k (r_{t+1} + \gamma \max_{a \in A} Q_{k-1}(s_{t+1}, a) - Q_{k-1}(s_t, a_t))$$

   3.3 Update Policy. $\pi = \pi_{Q_k}$.

This is an **off-policy** algorithm since we update the estimate of $Q_*$ with samples from other policies $\pi_{Q_k}$.
Problems of Q-Learning

Though quite elegant, Q-learning has some implementation problems which we have already faced before.

**Problems**

1. Computationally these algorithms are hard even in tabular case if number of states is large.
2. With infinite number of states implementation of Q-Learning procedure is not obvious.

So we need function approximation! This is why have appeared Fitted Q-iteration Riedmiller (2005) and DQN Mnih et al. (2015).
Take-home Messages & Future Works: Gradient Methods

We have considered the methods based on value functions (value-based). Could we directly model the policy instead?

▶ Assume $\pi_\theta : S \rightarrow \mathcal{P}(A)$ to be from some parametric family $\Pi_\theta$ (neural networks, for instance). If only we could compute

$$\nabla_\theta J^{\pi_\theta} = \nabla_\theta \mathbb{E} \left[ \sum_{t=0}^{\infty} \gamma^t R_{t+1} \right]$$

we would use gradient descent to train the policy!

▶ We could not get rid of expectation and the resulting gradient estimate has immense variance, so we need variance reduction...

▶ More on that in Maxim Kaledin’s course.
Take-home Messages & Future Works: Distributional Analysis of Q-learning and TD methods

- If we set stepsize $\alpha_k = \text{const}$, we still can prove convergence in expectation, but ....

- the iterations do not converge to a number but to some random variable! (Picture: (Sutton and Barto, 2018, Ex.6.6,p.132))

- The properties of this limiting distribution are investigated in Amortila et al. (2020) and some problems related to this are described in one of our projects.
RL theory virtual Seminar

https://sites.google.com/view/rltheoryseminars/home
RL is very interesting

RL is not an easy trip

Need to learn Markov chains (read my lecture notes + Eric Moulines’ book), stochastic approximation, dynamic programming etc

Follow mini-courses on stochastic approximation, multi-agents, gradient methods in RL

We are ready for ’real’ projects

Thank you!


