# Preliminaries needed to understand Proximal Policy Optimization Algorithms

Notes on discussion and derivations from Sutton's book and John Schulman's articles

D. Gueorguiev 12/13/23

## Table of Contents

## A bit of theory on Policy Gradient Reinforcement Learning Methods

Assumptions:

The environment can be represented by a finite MDP

This is equivalent of saying that its state $\mathcal{S}$, action $\mathcal{A}$ and reward $\mathcal{R}$ sets are finite, and its dynamics is given by a set of probabilities $p(s', r|s, a)$, for all $s \in \mathcal{S}, a \in \mathcal{A}(s), r \in \mathcal{R}$ and $s' \in \mathcal{S}^+$ ($\mathcal{S}^+$ is $\mathcal{S}$ plus a terminal state if the problem is episodic).

We would like to compute value functions to organize and search for good policies.

The optimal value functions satisfying the Bellman's optimality equations were derived and discussed in (Gueorguiev, 2023) (see Eq. (22) and (23)).

$$v_*(s) = \max_a \mathbb{E}\left[R_{t+1} + \gamma v_*(S_{t+1}) \mid S_t = s, A_t = a\right] = \max_a \sum_{s',r} p(s', r|s, a)[r + \gamma v_*(s')] \quad (1)$$

$$q_*(s, a) = \mathbb{E}\left[R_{t+1} + \gamma \max_{a'} q_*(S_{t+1}, a') \mid S_t = s, A_t = a\right] = \sum_{s',r} p(s', r|s, a)\left[r + \gamma \max_{a'} q_*(s', a')\right] \quad (2)$$

### Policy evaluation (Prediction)

#### *Policy evaluation for the state values*

**Definition**: *policy evaluation for state values* - computation of the state-value function $v_\pi$ for a given policy $\pi$. This is also known as the *prediction problem for state values*.

Question: How to compute the state-value function $v_\pi$ for an arbitrary policy $\pi$.

From (11) and (12) in (Gueorguiev, 2023) we can write:

$$
\begin{aligned}
v_\pi(s) &\doteq \mathbb{E}_\pi[G_t | S_t = s] \\
&= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} | S_t = s] && (3) \\
&= \mathbb{E}_\pi[R_{t+1} + \gamma v_\pi(S_{t+1}) | S_t = s] && (4) \\
&= \textstyle\sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)[r + \gamma v_\pi(s')] && (5)
\end{aligned}
$$

where $\pi(a|s)$ is the probability of taking action $a$ in state $s$ under policy $\pi$, and the expectations are subscribed by $\pi$ to indicate that they are conditional on $\pi$ being followed.
The existence and uniqueness of $v_\pi$ are guaranteed as long as either $\gamma < 1$ or eventual termination is guaranteed from all states under the policy $\pi$.
If the environment's dynamics are completely known, then (5) is a system of $|\mathcal{S}|$ simultaneous linear equations in $|\mathcal{S}|$ unknowns (the $v_\pi(s), s \in \mathcal{S}$).

Clearly, $v_k = v_\pi$ is fixed point for (5) because the Bellman equation for $v_\pi$ assures equality in this case. We are going to be looking into iterative solution of (5). Indeed, the sequence $\{v_k\}$ can be shown to converge to $v_\pi$ under the same conditions which guarantee the existence of $v_\pi$. This algorithm is known as *iterative policy evaluation*. To produce each successive approximation , the iterative policy evaluation applies the same operation to each state $s$: it replaces the old value of $s$ with a new value obtained from the old values of the successor states of $s$, along all the one-step transitions possible under the policy being evaluated. We call this kind of operation an *expected update*. Each iteration of the iterative policy evaluation updates the value of every state once to produce the new approximate value function $v_{k+1}$.

Note: There are several different kinds of expected updates, depending on whether $a$) a **state** (as in (5)) or $b$) a **state-action pair** is being updated, and, depending on the precise way the estimated values of the successor states are combined.

Note2: All the updates done in the algorithms based on Bellman equations are ***expected updates*** because they are based on the expectation over ***all possible next states*** rather than on a sample next state.

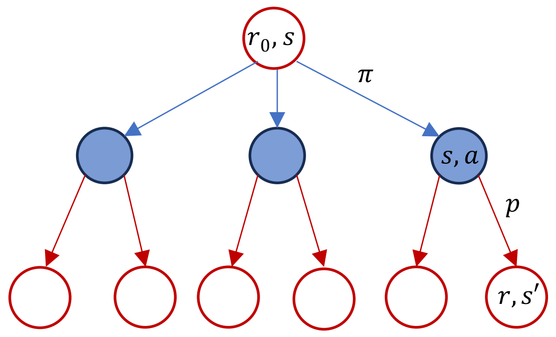
Note3: the nature of the update can be expressed by using backup diagram (backup diagrams were discussed in (Gueorguiev, 2023) and in Chapter 3 of (Richard D. Sutton, 2020)).

In-place Algorithm for iterative policy evaluation:
Input: Policy $\pi$ to be evaluated, the environment dynamics $p$
Output: $V \approx v_\pi$
Algorithm Parameter: $\theta > 0$, small threshold determining accuracy of estimation

Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(terminal) = 0$.

```
Loop:
  $\Delta \leftarrow 0$
  Loop for each $s \in \mathcal{S}$:
    $v \leftarrow V(s)$
    $V(s) \leftarrow \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)[r + \gamma V(s')]$
    $\Delta \leftarrow max(\Delta, |v - V(s)|)$
Until $\Delta < \theta$
```

Figure 1: Backup diagram for the update relation (5)



Note: this is the same backup diagram on Figure 1 in (Gueorguiev, 2023).

*Short explanation on the backup diagram above taken from* (Gueorguiev, 2023)*:*
Each open circle, which will be denoted as *reward-state node* so forth, colored in red represents a state and the reward, which is associated with this state  For instance, the root node shown on Figure 1 has associated reward $r_0$ and state $s$. Each solid circle, colored in blue represents a state-action pair and will be denoted as *state-action node* so forth. The specific state on the rightmost state-action node is shown as $(s, a)$.  Each directed blue edge connects state node with state-action node and represents application of the policy $\pi$ to the root reward-state node $(r_0, s)$.  Each directed red edge emanating from a state-action node ends in a possible reward-state node corresponding to specific probable pair of reward $r$ and new state $s'$. Thus, each directed red edge represents the application of the function $p$ of the MDP dynamics. The Bellman equation (5) averages over all of the possibilities weighing each possibility represented by a path from the root of the Backup diagram on Figure 1 to a leaf by its probability of occurring. It states that the value of the start state must equal the discounted value of the expected next state plus the reward expected along the way.

## *Policy evaluation for the state-action values*

An algorithm on iterative policy evaluation for the state-action values can be constructed using the Bellman equations for state-action values.

Starting from Bellman equation for state-action values we write:

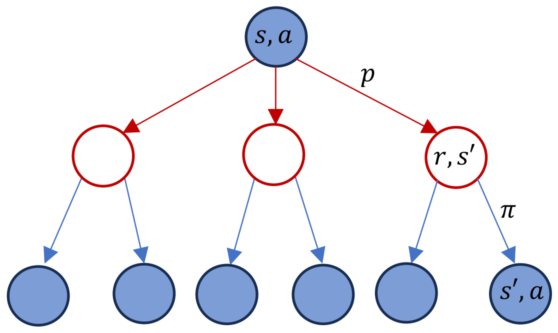
In Chapter 3 of (Richard D. Sutton, 2020) $q_\pi(s, a)$ is defined as:
$q_\pi(s, a) = \mathbb{E}_\pi[G_t | S_t = s, A_t = a]$     (6)
(also see Eq (7) in (Gueorguiev, 2023))

Using the recursive relation for the total return $G_t$ in the right-hand side of (6) gives:
$\mathbb{E}_\pi[G_t | S_t = s, A_t = a] = \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} | S_t = s, A_t = a]$     (7)

The expectation in the right-hand side of (7) is explicitly expressed in terms of the environment dynamics
$\mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} | S_t = s, A_t = a] = \sum_{s', r} p(s', r | s, a)[r + \gamma q_\pi(s', a')]$    (8)

Thus we arrive at the Bellman equation for the state-action values:
$q_\pi(s, a) = \sum_{s', r} p(s', r | s, a)[r + \gamma q_\pi(s', a')]$     (9)

Figure 2: Backup diagram for the update relation (9)



//TODO: finish the algorithm of policy-evaluation for state-action values

## Policy Improvement

We are computing the (state/state-action) value function for a policy to help find better policies.
Let us assume that we have determined the value function $v_\pi$ for arbitrary deterministic policy $\pi$. For some state $s$ we would like to know whether or not we should change the policy to deterministically choose an action $a \neq \pi(s)$.
We know how good it is to follow the current policy from $s$ – that is, $v_\pi(s)$, is the expected return when starting in $s$ and following $\pi$ thereafter.
Question: would be for a better or worse to change to a new policy from the current moment on (that is, state $s$) ?
One way to answer this question is to consider selecting $a$ in $s$ and thereafter following the existing policy $\pi$.
The value of this way of behaving is:

$$\begin{aligned} q_\pi(s,a) &\doteq \mathbb{E}[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s, A_t = a] \\ &= \sum_{s',r} p(s',r|s,a)[r + \gamma v_\pi(s')] \qquad (10) \end{aligned}$$

The key criterion is whether $q_\pi(s,a)$ computed by (10) is greater or less than $v_\pi(s)$. If it is greater this means it is better to select $a$ once in $s$, and thereafter follow $\pi$. In such case one would expect that it would be better still to select $a$ every time $s$ is encountered than it would be to follow $\pi$ all the time. So one would expect that the new policy $\pi'$ constructed in such way by amending $\pi$ would be better one overall. This is a special case of the *policy improvement theorem*.

**Policy improvement theorem**
Let $\pi$ and $\pi'$ be any pair of deterministic policies such as that, for all $s \in \mathcal{S}$,

$$q_\pi(s, \pi'(s)) \geq v_\pi(s) \qquad (11)$$

Then the policy $\pi'$ must be as good as, or better than, $\pi$. That is, it must obtain greater or equal expected return from all states $s \in \mathcal{S}$:

$$v_{\pi'}(s) \geq v_\pi(s) \qquad (12)$$

Moreover, if there is strict inequality of (11) at any state $s$, then there must be strict inequality of (12) at the same state $s$.

The policy improvement theorem applies to the two policies which we considered in the earlier paragraph – the original deterministic policy $\pi$ and a changed policy $\pi'$ that is identical to $\pi$ except that $\pi'(s) = a \neq \pi(s)$. For states other than s, (11) holds because the two sides are equal. Thus, if $q_\pi(s,a) > v_\pi(s)$ then $\pi'$ is better than $\pi$.

<u>Idea of proof</u>:

From (11) we have:

$$\begin{aligned} v_\pi(s) &\le q_\pi(s, \pi'(s)) \\ &= \mathbb{E}[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s, A_t = \pi'(s)] \quad \text{(by the definition of } q_\pi(s,a)\text{)} \\ &= \mathbb{E}_{\pi'}[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s] \\ &\le \mathbb{E}_{\pi'}[R_{t+1} + \gamma q_\pi(S_{t+1}, \pi'(S_{t+1})) \mid S_t = s] \quad \text{(by (11))} \\ &= \mathbb{E}_{\pi'}[\quad] \end{aligned}$$

## Appendix

### Solution of the Bellman system of equations for state values

We notice that the Bellman system of equations with respect to $v_\pi$ (5) can be rewritten as:

$$v_\pi(s_i) = \sum_a \pi(a|s_i) \sum_{j\neq i,r} p(s_j, r|s_i, a)[r + \gamma v_\pi(s_j)] + \sum_a \pi(a|s_i) \sum_r p(s_i, r|s_i, a)[r + \gamma v_\pi(s_i)] \quad \text{(A1)}$$

hence

$$v_\pi(s_i) - \gamma \sum_a \pi(a|s_i) \sum_r p(s_i, r|s_i, a) \cdot v_\pi(s_i) - \gamma \sum_a \pi(a|s_i) \sum_{j\neq i,r} p(s_j, r|s_i, a) \cdot v_\pi(s_j)$$
$$= \sum_a \pi(a|s_i) \sum_r p(s_i, r|s_i, a) \cdot r + \sum_a \pi(a|s_i) \sum_r p(s_i, r|s_i, a) \cdot r \quad \text{(A2)}$$

The left-hand side of (A2) can be rewritten as:

$$v_\pi(s_i) - \gamma \sum_r \sum_a p(s_i, r|s_i, a) \cdot \pi(a|s_i)\, v_\pi(s_i) - \gamma \sum_{j\neq i,r} \sum_a p(s_j, r|s_i, a) \cdot \pi(a|s_i)\, v_\pi(s_j) \quad \text{(A3)}$$

The right-hand side of (A2) are rearranged as :

$$\sum_r \sum_a p(s_i, r|s_i, a) \cdot \pi(a|s_i) \cdot r + \sum_{j\neq i,r} \sum_a p(s_j, r|s_i, a) \cdot \pi(a|s_i) \cdot r \quad \text{(A4)}$$

we denote with $p_\pi(s_j, r|s_i)$, $p_\pi(s_j|s_i)$ and $r_\pi(s_j|s_i)$ the following expressions:

$$p_\pi(s_j, r|s_i) \doteq \mathbb{E}[\pi | S_{t-1} = s_i] = \sum_a p(s_j, r|s_i, a) \cdot \pi(a|s_i) \quad \text{(A5)}$$

$$p_\pi(s_j|s_i) \doteq \sum_r p_\pi(s_j, r|s_i) \quad \text{(A6)}$$

$$r_\pi(s_j|s_i) \doteq \sum_r p_\pi(s_j, r|s_i) \cdot r \quad \text{(A7)}$$

Using (A3)-(A7) in (A2) leads to :

$$(1 - \gamma p_\pi(s_i|s_i)) v_\pi(s_i) - \gamma \sum_{j\neq i} p_\pi(s_j|s_i) v_\pi(s_j) = r_\pi(s_i|s_i) + \sum_{j\neq i} r_\pi(s_j|s_i) \quad \text{(A8)}$$

(A8) represents a linear system of equations with respect to the $|\mathcal{S}|$ unknowns $v_\pi(s), s \in \mathcal{S}$.

(A8) in matrix form:

$$A \cdot v_\pi(s) = b \quad \text{(A9)}$$

For convenience we abbreviate:

$$p_{j,i} \doteq p_\pi(s_j|s_i)\ ,\ r_i \doteq \sum_j r_\pi(s_j|s_i), S \doteq |\mathcal{S}| \qquad \text{(A10)}$$

Thus,

$$A = \begin{bmatrix} 1-\gamma p_{11} & p_{21} & \cdots & p_{S1} \\ p_{12} & 1-\gamma p_{22} & & p_{S2} \\ \vdots & & \ddots & \vdots \\ p_{1S} & p_{2S} & \cdots & 1-\gamma p_{SS} \end{bmatrix},\quad b = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_S \end{bmatrix} \qquad \text{(A11)}$$

//TODO: derive degeneracy conditions on the function of the environment dynamics

## Bibliography

Gueorguiev, D. (2023, Nov 26). *Note on Q functions and V functions in Reinforcement Learning.* Retrieved from Reinforcement Learning and Game Theory Repository: https://github.com/dimitarpg13/reinforcement_learning_and_game_theory/blob/main/docs/Note_on_Q_functions_in_Reinforcement_Learning.pdf

Heeswijk, W. v. (2021, May 11). *The Four Policies of Reinforcement Learning.* Retrieved from https://github.com/dimitarpg13/reinforcement_learning_and_game_theory/blob/main/articles/ReinforcementLearning/The_Four_Policy_Classes_of_Reinforcement_Learning_Wouter_van_Heeswijk_TDS.pdf

Heeswijk, W. v. (2022, Apr 9). *Policy Gradients In Reinforcement Learning Explained.* Retrieved from Towards Data Science: https://github.com/dimitarpg13/reinforcement_learning_and_game_theory/blob/main/articles/ReinforcementLearning/Policy_Gradients_In_Reinforcement_Learning_Explained_Wouter_van_Heeswijk_TDS.pdf

Heeswijk, W. v. (2022, Nov 29). *Proximal Policy Optimization (PPO) Explained.* Retrieved from Towards Data Science: https://github.com/dimitarpg13/reinforcement_learning_and_game_theory/blob/main/articles/ReinforcementLearning/Proximal_Policy_Optimization_Algorithms_Shulman_2017.pdf

John Schulman, e. a. (2018, Oct 20). *High-Dimensional Continuous Control Using Generalized Advantage Estimation.* Retrieved from arxiv.org: https://github.com/dimitarpg13/reinforcement_learning_and_game_theory/blob/main/articles/ReinforcementLearning/High-Dimensional_Continuous_Control_Using_Generalized_Advantage_Estimation_Schulman_2018.pdf

John Schulman, F. W. (2017, Aug 28). *Proximal Policy Optimization Algorithms.* Retrieved from arxiv.org: https://github.com/dimitarpg13/reinforcement_learning_and_game_theory/blob/main/articles/ReinforcementLearning/Proximal_Policy_Optimization_Algorithms_Shulman_2017.pdf

John Schulman, S. L. (2017, Apr 20). *Trust Region Policy Optimization.* Retrieved from arxiv.org: https://github.com/dimitarpg13/reinforcement_learning_and_game_theory/blob/main/articles/ReinforcementLearning/TrustRegionPolicyOptimization_Schulman_2015.pdf

OpenAI. (2017, July 20). *openai baselines proximal policy optimization.* Retrieved from openai.com: https://openai.com/research/openai-baselines-ppo

Richard D. Sutton, A. G. (2020, Oct 12). *Reinforcement Learning: An Introduction, Second Edition.* Retrieved from https://github.com/dimitarpg13/reinforcement_learning_and_game_theory/blob/main/books/ReinforcementLearningSuttonSecondEdition2020.pdf

Sham Kakade, J. L. (2002, July 8). *Approximately Optimal Approximate Reinforcement Learning.* Retrieved from ICML.cc: https://github.com/dimitarpg13/reinforcement_learning_and_game_theory/blob/main/articles/ReinforcementLearning/Approximate_Optimal_Approximate_Reinforcement_Learning_KakadeLangford-icml2002.pdf

Volodymir Mnih, e. a. (2016, June 16). *Asynchronour Methods for Deep Reinforcement Learning.* Retrieved from https://github.com/dimitarpg13/reinforcement_learning_and_game_theory/blob/main/articles/ReinforcementLearning/Asynchronous_Methods_for_Deep_Reinforcement_Learning_Mnih_2016.pdf

Volodymyr Mnih, e. a. (2015, Feb 26). *Human-level control through deep reinforcement learning.* Retrieved from Nature.com: https://github.com/dimitarpg13/reinforcement_learning_and_game_theory/blob/main/articles/ReinforcementLearning/Human-level_control_through_deep_reinforcement_learning_Mnih_2015.pdf