Reinforcement Learning
Reinforcement Learning: Policy Approach

E. Le Pennec

École Polytechnique

M2 DS - Fall 2022
Outline

- Operations Research and MDP.
- Reinforcement learning and interactions.
- More tabular reinforcement learning.
- Reinforcement and approximation of value functions.
- Actor/Critic: a Policy Point of View
How to find the best policy knowing the MDP?

- Is there an optimal policy?
- How to estimate it numerically?

- Finite states/actions space assumption (tabular setting).
- Focus on interactive methods using value functions (dynamic programming).
- Policy deduced by a statewise optimization of the value function over the actions.
- Most results for the discounted setting.
Reinforcement Learning and Interactions

How to find the best policy not knowing the MDP?

- How to interact with the environment to learn a good policy?
- Can we use a Monte Carlo strategy outside the episodic setting?
- How to update value functions after each interaction?

- Focus on stochastic methods using tabular value functions ($Q$ learning, SARSA...)
- Policy deduced by a statewise optimization of the value function over the actions.
More Tabular Reinforcement Learning

Can We Do Better?

- Is there a gain to wait more than one step before updating?
- Can we interact with a different policy than the one we are estimating?
- Can we use an estimated model to plan?
- Can we plan in real time instead of having to do it beforehand?

- Finite states/actions space setting (tabular setting).
How to Deal with a Large/Infinite states/action space?

- How to approximate value functions?
- How to estimate good approximation of value functions?

- Finite action space setting.
- Stochastic algorithm (Deep Q Learning...).
- Policy deduced by a statewise optimization of the value function over the actions.
Could We Directly Parameterized the Policy?

- How to parameterize a policy?
- How to optimize this policy?
- Can we combine parametric policy and approximated value function?

- State Of The Art Algorithms (DPG, PPO, SAC...)

Actor/Critic: a Policy Point of View
Outline

1. Policy Gradient Theorems
2. Monte Carlo Based Policy Gradient
3. Actor / Critic Principle
4. 3 SOTA Algorithms
5. Average Return
6. References
Policy Point of View

Optimize policy directly instead of deriving it from a value function.
Avoid the argmax operator.
Most natural POV?

Pontryagin vs Hamilton-Jacobi(-Bellman) in control!
Outline

1 Policy Gradient Theorems
2 Monte Carlo Based Policy Gradient
3 Actor / Critic Principle
4 3 SOTA Algorithms
5 Average Return
6 References
Policy and Goal

\[ J_\mu(\pi) = \sum_s \mu(s)v_\pi(s) \]

Goal: average expected return over the states

- Target used to define the linear programming formulation of an optimal policy in the tabular setting.
- \( \mu \) can be the initial distribution of the states (independent of \( \pi \))...
- but may also depends on \( \pi \) (for instance the associated stationary measure)
- Other choices will appear.

- Goal: optimize \( J_\mu(\pi) \) in \( \pi \)!
Parametric Policy

\[
\pi_\theta(a|s) = \begin{cases} 
\frac{e^{h_\theta(a,s)}}{\sum_{a'} e^{h_\theta(a',s')}} & \text{(softmax)} \\
P_{h_\theta(s)}(a) & \text{(parametric conditional model)} \\
1_{a=h_\theta(s)} & \text{(deterministic)}
\end{cases}
\]

**Parametric Policy**

- Restriction of the set of policy to a parametrized one.
- Most classical parametrizations:
  - Soft-max with a preference function \( h_\theta(a,s) \),
  - Parametric conditional model with parameter \( h_\theta(s) \)
- To be useful need to be able to sample the distribution.
- \( h_\theta \): from linear model to deep learning . . .
- Most of our result will assume that \( \pi_\theta(a|s) \) is differentiable with respect to \( \theta \).
- Deterministic policies will be considered with a different analysis.
Episodic Setting: Gradient of Expected Returns

\[ v_{\pi_\theta}(s) = \mathbb{E}_{\pi_\theta}[G_0|S_0 = s] \]

\[ \nabla_\theta v_{\pi_\theta}(s) = \mathbb{E}_{\pi_\theta} \left[ \left( \sum_{t=0}^{T_\tau - 1} \nabla \log \pi_\theta(A_t|S_t) \right) G_0 \bigg| S_0 = s \right] \]

Expected Returns

- Rely on \( v_{\pi_\theta}(s) = \sum_\tau \mathbb{P}_{\pi_\theta}(\tau|S_0 = s) G_0(\tau) \) and

\[ \nabla \mathbb{P}_{\pi_\theta}(\tau|S_0 = s) = \mathbb{P}_{\pi_\theta}(\tau|S_0 = s) \nabla \log \mathbb{P}_{\pi_\theta}(\tau|S_0 = s) \]

\[ = \mathbb{P}_{\pi_\theta}(\tau|S_0 = s) \sum_t \left( \nabla \log \pi_\theta(A_t|S_t) + \nabla p(R_{t+1}, S_{t+1}|S_t, A_t) \right) \]

\[ = \mathbb{P}_{\pi_\theta}(\tau|S_0 = s) \sum_t \nabla \log \pi_\theta(A_t|S_t) \]

- In an episodic setting, any trajectory \( \tau \) ends at a finite time \( T_\tau \).
Episodic Setting: Policy Gradient Theorem

\[ J_{\mu_0}(\pi_\theta) = \sum_s \mathbb{P}(S_0 = s) v_{\pi_\theta}(s) \]

\[ \nabla J_{\mu_0}(\pi_\theta) = \mathbb{E}_{\pi_\theta} \left[ \left( \sum_{t=0}^{T_{\tau}-1} \nabla \log \pi_\theta(A_t | S_t) \right) G_0 \right] \]

Policy Gradient Theorem

- Natural \( \mu \): initial state distribution.
- Gradient is an expectation: MC type algorithm...
- Can be interpreted as the gradient of a the maximum likelihood of the actions weighted by the return.
- Favors good actions over bad ones.
Baseline and Variance Reduction

\[ J_{\mu_0}(\pi_\theta) = \sum_s \mathbb{P}(S_0 = s) \nu_{\pi_\theta}(s) \]

\[ \nabla J_{\mu_0}(\pi_\theta) = \mathbb{E}_{\pi_\theta} \left[ \left( \sum_{t=0}^{T_\tau-1} \nabla \log \pi_\theta(A_t|S_t) \right) (G_0 - b) \right] \]

Variance Reduction and Baseline

- The previous formulae are valid if one replace \( G_0 \) by any function of \( \tau \).
- For any constant \( b \), this leads to
  \[ \nabla \mathbb{E}_{\pi_\theta}[b] = 0 = \mathbb{E}_{\pi_\theta} \left[ \left( \sum_{t=0}^{T_\tau-1} \nabla \log \pi_\theta(A_t|S_t) \right) b \right] \]
- Optimal value for
  \[ b = \mathbb{E}_{\pi_\theta} \left[ \left( \sum_{t=0}^{T_\tau-1} \nabla \log \pi_\theta(A_t|S_t) \right)^2 G_0 \right] / \mathbb{E}_{\pi_\theta} \left[ \left( \sum_{t=0}^{T_\tau-1} \nabla \log \pi_\theta(A_t|S_t) \right)^2 \right] \]
- Most used value \( b = \mathbb{E}_{\pi_\theta}[G_0] \).
Episodic/Discounted: Gradient(s) of Expected Return

\[ v_{\pi_\theta}(s) = \mathbb{E}_{\pi_\theta} \left[ \sum \gamma^t R_t \bigg| S_0 = s \right] \]

\[ \nabla v_{\pi_\theta}(s) = \sum_t \gamma^t \mathbb{E}_{\pi_\theta} \left[ \left( \sum_{t'=0}^{t-1} \nabla \log \pi_\theta(A_{t'}|S_{t'}) \right) R_t \bigg| S_0 = s \right] \]

\[ = \sum_{t'} \mathbb{E}_{\pi_\theta} \left[ \nabla \log \pi_\theta(A_{t'}|S_{t'}) \left( \sum_{t \geq t'} \gamma^t R_t \right) \bigg| S_0 = s \right] \]

\[ = \sum_{t'} \gamma^{t'} \mathbb{E}_{\pi_\theta} \left[ \nabla \log \pi_\theta(A_{t'}|S_{t'}) Q_{\pi_\theta}(S_{t'}, A_{t'}) \big| S_0 = s \right] \]

\[ = \sum_{t'} \gamma^{t'} \mathbb{E}_{\pi_\theta} \left[ \nabla \log \pi_\theta(A_{t'}|S_{t'}) \left( Q_{\pi_\theta}(S_{t'}, A_{t'}) - V_{\pi_\theta}(S_{t'}) \right) \bigg| S_0 = s \right] \]

**Expected Returns**

- Several gradients!
Episodic/Discounted: More Gradient(s)

\[ \nabla v_{\pi_\theta}(s) = \sum_{t'} \gamma^{t'} \mathbb{E}_{\pi_\theta} [\nabla \log \pi_\theta(A_{t'}|S_{t'}) Q_{\pi_\theta}(S_{t'}, A_{t'}) | S_0 = s] \\
= \sum_{t'} \gamma^{t'} \mathbb{E}_{\pi_\theta} [\nabla \log \pi_\theta(A_{t'}|S_{t'}) A_{\pi_\theta}(S_{t'}, A_{t'}) | S_0 = s] \\
= \sum_s \left( \sum_t \gamma^t \mathbb{P}_{\pi_\theta}(S_t = s | S_0 = s) \right) \left( \sum_a \pi_\theta(a|s) \nabla \log \pi_\theta(a|s) Q_{\pi_\theta}(s,a) \right) \\
= \sum_s \left( \sum_t \gamma^t \mathbb{P}_{\pi_\theta}(S_t = s | S_0 = s) \right) \left( \sum_a \pi_\theta(a|s) \nabla \log \pi_\theta(a|s) A_{\pi_\theta}(s,a) \right) \\

Focus on states
- More gradients!
Episodic/Discounted: Policy Gradient(s)

\[ J(\mu_0) (\pi_\theta) = \sum_s \mu_0(s) v_{\pi_\theta}(s) \]

\[ \nabla J_{\mu_0}(\pi_\theta) = \sum_s \left( \sum_t \gamma^t P_{\pi_\theta}(S_t = s) \right) \left( \sum_a \pi_\theta(a|s) \nabla \log \pi_\theta(a|s) Q_{\pi_\theta}(s, a) \right) \]

\[ = \sum_s \left( \sum_t \gamma^t P_{\pi_\theta}(S_t = s) \right) \left( \sum_a \pi_\theta(a|s) \nabla \log \pi_\theta(a|s)(Q_{\pi_\theta}(s, a) - V_{\pi_\theta}(s, a)) \right) \]

**Discounted Setting**

- Average (discounted) return from the beginning.
- Focus on early steps in discounted setting...
Policy Improvement Lemma

\[ J_{\mu_0}(\pi') - J_{\mu_0}(\pi) = \sum_t \gamma^t P_{\pi'}(S_t = s) \left( \sum_a (\pi'(a|s) - \pi(a|s)) Q_\pi(s, a) \right) \]

Proof in the discounted setting rely on

\[ v_{\pi'} - v_\pi = r_{\pi'} + \gamma P_{\pi'} v_{\pi'} - r_\pi - \gamma P_\pi v_\pi \]

\[ = r_{\pi'} - r_\pi + \gamma (P_{\pi'} - P_\pi) v_\pi + \gamma P_{\pi'} (v_{\pi'} - v_\pi) \]

\[ v_{\pi'} - v_\pi = (I - \gamma P_{\pi'})^{-1} \left( r_{\pi'} - r_\pi + \gamma (P_{\pi'} - P_\pi) v_\pi \right) \]
Approximate Policy Improvement Lemma

\[
\left| J_{\mu_0}(\pi') - J_{\mu_0}(\pi) - \sum_t \gamma^t P_\pi(S_t = s) \left( \sum_a (\pi'(a|s) - \pi(a|s)) A_\pi(s, a) \right) \right|
\]

\[
= \left| \sum_t \gamma^t (P_{\pi'}(S_t = s) - P_\pi(S_t = s)) \left( \sum_a (\pi'(a|s) - \pi(a|s)) A_\pi(s, a) \right) \right|
\]

\[
\leq \sum_t \gamma^t 2t \max_s ||\pi'(\cdot|s) - \pi(\cdot|s)||_1^2 \max_{s,a} |A_\pi(s, a)|
\]

Approximate Policy Improvement Lemma

- If \( \max_s ||\pi'(\cdot|s) - \pi(\cdot|s)||_1 \leq \epsilon \)
  \[P_{\pi'}(S_t = s) = (1 - \epsilon)^t P_\pi(S_t = s) + (1 - (1 - \epsilon)^t) P_{\text{mistake}}(S_t = s)\]
  \[\rightarrow |P_{\pi'}(S_t = s) - P_\pi(S_t = s)| \leq 2(1 - (1 - \epsilon)^t) \leq 2\epsilon t\]
Approximate Policy Improvement Lemma

\[ J_{\mu_0}(\pi') - J_{\mu_0}(\pi) - \sum_t \gamma^t \mathbb{P}_\pi(S_t = s) \left( \sum_a (\pi'(a|s) - \pi(a|s)) A_\pi(s, a) \right) \]
\[ \leq \sum_t \gamma^t 2t \max_s \| \pi'(\cdot|s) - \pi(\cdot|s) \|_1^2 \max_{s,a} |A_\pi(s, a)| \]

Approximate Policy Improvement Lemma and Policy Gradient Theorem

- Let \( \pi' = \pi_{\theta+h} \) and \( \pi_\theta \)
  - \( \pi_{\theta+h}(a|s) - \pi_\theta(a|s) = \pi_\theta(a|s) \langle \nabla \log \pi_\theta(a|s), h \rangle + O(\|h\|^2) \)
  - \( \| \pi_{\theta+h}(\cdot|s) - \pi_\theta(\cdot|s) \|_1 \leq \|h\| \max_a \| \nabla \log \pi_\theta(a|s) \| + O(\|h\|^2) \)
- Implies Policy Gradient Theorem:
  \[ J_{\mu_0}(\pi_{\theta+h}) \]
  \[ = J_{\mu_0}(\pi_\theta) + \sum_t \gamma^t \mathbb{P}_{\pi_\theta}(S_t = s) \left( \sum_a \pi_\theta(a|s) \langle \nabla \log \pi_\theta(s, a), h \rangle A_\pi(s, a) \right) + O(\|h\|^2) \]
Outline

1. Policy Gradient Theorems
2. Monte Carlo Based Policy Gradient
3. Actor / Critic Principle
4. 3 SOTA Algorithms
5. Average Return
6. References
Monte Carlo Approach

Monte Carlo Based Policy
Gradient

Monte Carlo

Replace every return by an empirical estimate along episodes.

Need to wait until the end of the episodes.

\[ G_t = \sum_{t' \geq t} R_{t+1} \]

\[ Q_{t,\pi_\theta}(s, a) = \mathbb{E}[G_t | S_t = s, A_t = a] \]
REINFORCE: Monte Carlo Based Policy Gradient

\[ J_{\mu_0}(\pi_\theta) = \sum_s P(S_0 = s) v_{\pi_\theta}(s) \]

\[ \nabla J_{\mu_0}(\pi_\theta) = \mathbb{E}_{\pi_\theta} \left[ \left( \sum_{t=0}^{T-1} \nabla \log \pi_\theta(A_t|S_t) \right) G_0 \right] \]

\[ = \sum_s \left( \sum_t P_{\pi_\theta}(S_t = s) \right) \left( \sum_a \pi_\theta(a|s) \nabla \log \pi_\theta(a|s) Q_{\pi_\theta}(s, a) \right) \]

\[ \hat{\nabla} J_{\mu_0}(\pi_\theta) = \left( \sum_{t=0}^{T-1} \nabla \log \pi_\theta(A_t|S_t) \right) G_0 \quad \text{or} \quad \hat{\nabla} J_{\mu_0}(\pi_\theta) = \sum_t \nabla \log \pi_\theta(A_t|S_t) G_t \]

REINFORCE

- Plain MC (SGD) algorithm.
- Need to wait until the end of the episods.
- Convergence guarantees (even in off-line setting with importance sampling).
Monte Carlo Based Policy Gradient

REINFORCE with Baseline

\[ \nabla J_{\mu_0}(\pi_\theta) = \mathbb{E}_{\pi_\theta} \left[ \left( \sum_{t=0}^{T-1} \nabla \log \pi_\theta(A_t|S_t) \right) (G_0 - b) \right] \]

\[ = \sum_s \left( \sum_t \mathbb{P}_{\pi_\theta}(S_t = s) \right) \left( \sum_a \pi_\theta(a|s) \nabla \log \pi_\theta(a|s) (Q_{\pi_\theta}(s, a) - b(s)) \right) \]

\[ \hat{\nabla} J_{\mu_0}(\pi_\theta) = \left( \sum_{t=0}^{T-1} \nabla \log \pi_\theta(A_t|S_t) \right) (G_0 - b) \]

or \[ \hat{\nabla} J_{\mu_0}(\pi_\theta) = \sum_t \nabla \log \pi_\theta(A_t|S_t) (G_t - b(S_t)) \]

REINFORCE with baseline

- Several choices for \( b \)...
- and for \( b(s) \) which can be any function (a crude estimate of \( V_t,\pi(s) \) for instance)!
- Convergence guarantees (even in off-line setting with importance sampling).
Monte Carlo Based Policy
Gradient

Discounted REINFORCE?

\[
\nabla J_{\mu_0}(\pi_\theta) = \mathbb{E}_{\pi_\theta} \left[ \left( \sum_{t=0}^{T-1} \nabla \log \pi_\theta(A_t|S_t) \right) (G_0 - b) \right] \\
= \sum_s \left( \sum_t \gamma^t \mathbb{P}_{\pi_\theta}(S_t = s) \right) \left( \sum_a \pi_\theta(a|s) \nabla \log \pi_\theta(a|s) \left( Q_{\pi_\theta}(s, a) - b(s) \right) \right) \\
\hat{\nabla} J_{\mu_0}(\pi_\theta) = \left( \sum_{t=0}^{T-1} \nabla \log \pi_\theta(A_t|S_t) \right) (G_0 - b) \\
\text{or} \quad \hat{\nabla} J_{\mu_0}(\pi_\theta) = \sum_t \gamma^t \nabla \log \pi_\theta(A_t|S_t) (G_t - b(S_t))
\]

Discounted REINFORCE

- Can be defined...
- but still requires an episodic setting for the discounted return $G_t$ to be computed.
Monte Carlo Based Policy Gradient

\[ \hat{\nabla} J_{\mu_0}(\pi_\theta) = \sum_t \gamma^t \nabla \log \pi_\theta(A_t|S_t)(G_t - b(S_t)) \]

\[ \rightarrow \hat{\nabla} J_{\mu_\pi_\theta}(\pi_\theta) = \frac{1}{1 - \gamma} \nabla \log \pi_\theta(A_t|S_t)(G_t - b(S_t)) \]

Discounted Measure?

- Much less weights for later states!
- Probability independent of \( t \) if the initial distribution is the stationary distribution \( \mu_\pi_\theta \) corresponding to \( \pi_\theta \) (if it exists).
- Approximately true after a burning stage if we reach stationarity...
- Better handled by the average return!

- More on this later...
Outline

1. Policy Gradient Theorems
2. Monte Carlo Based Policy Gradient
3. Actor / Critic Principle
4. 3 SOTA Algorithms
5. Average Return
6. References
Actor / Critic

- **Actor**: Parametric policy $\pi_\theta$ used.
- **Critic**: $Q$-value function $Q_w(\cdot, \cdot)$ approximating $Q_{\pi_\theta}$.
- **Critic follows the Actor**, which is optimized using the Critic.

- In Value Approximation, the Actor follows the Critic (through the argmax operator).
- In on-line methods, the Actor is used to interact with the environment.
Actor / Critic Principle

\[ J(\mu_0)(\pi_\theta) = \sum_s \mu_0(s) v_{\pi_\theta}(s) \]

\[ \nabla J_{\mu_0}(\pi_\theta) = \sum_s \left( \sum_t \gamma^t \mathbb{P}_{\pi_\theta}(S_t = s) \right) \left( \sum_a \pi_\theta(a|s) \nabla \log \pi_\theta(a|s) (Q_{\pi_\theta}(s, a) - V_{\pi_\theta}(s, a)) \right) \]

\[ \hat{\nabla} J_{\mu_0}(\pi_\theta) = \sum_t \gamma^t \pi_\theta(A_t|S_t) \nabla \log \pi_\theta(A_t|S_t) \left( Q_{\pi_\theta}(S_t, A_t) - \sum_a \pi(A_t|S_t) Q_{\pi_\theta}(S_t, A_t) \right) \]

\[ \simeq \sum_t \gamma^t \pi_\theta(A_t|S_t) \nabla \log \pi_\theta(A_t|S_t) \left( Q_{w}(S_t, A_t) - \sum_a \pi(A_t|S_t) Q_{w}(S_t, A_t) \right) \]

**Actor/Critic**

- Critic update: Stochastic Policy Gradient with plugin.
- Actor update: any Q-value methods estimating \( Q_{\pi_\theta} \).
- Requires a two scale algorithm so that \( Q_{w} \) is always a good estimate of \( Q_{\pi_\theta} \).

- Is this a real algorithm in a non episodic setting?
Actor / Critic

\[ J_{\mu_{\pi_\theta}}(\pi_\theta) = \sum_s \mu_{\pi_\theta}(s) v_{\pi_\theta}(s) \]

\[ \nabla J_{\mu_{\pi_\theta}}(\pi_\theta) = \sum_s \frac{1}{1 - \gamma} P_{\pi_\theta}(S_t = s) \left( \sum_a \pi_\theta(a|s) \nabla \log \pi_\theta(a|s) (Q_{\pi_\theta}(s, a) - V_{\pi_\theta}(s, a)) \right) \]

\[ \hat{\nabla} J_{\mu_{\pi_\theta}}(\pi_\theta) \approx \frac{1}{1 - \gamma} \pi_\theta(A_t|S_t) \nabla \log \pi_\theta(A_t|S_t) \left( Q_w(S_t, A_t) - \sum_a \pi(a|S_t) Q_w(S_t, A_t) \right) \]

**Actor / Critic**

- Critic update: Stochastic Policy Gradient with plugin.
- Actor update: any Q-value methods estimating \( Q_{\pi_\theta} \).
- Requires a two scale algorithm so that \( Q_w \) is always a good estimate of \( Q_{\pi_\theta} \).

- Require the existence of a stationary measure... and that this stationary measure is reached quickly.
- Much harder to do off-policy algorithm as the stationary measure is not known!
Critic in Actor/Critic

\[ Q_w \approx Q_{\pi_\theta} \]

**Critic**

- On-line TD learning with interaction following \( \pi_\theta \).
- Off-Policy TD learning is possible if the policy used for any action is stored.
- Approximate off-policy TD learning is possible using a replay buffer providing \( \pi_\theta \) is changing slowly.

- May lead to 3 scales algorithm (Actor/Critic Target/Critic)
- As mentioned in the previous slide, much harder to do off-line update for the actor.
Off-Line Actor

\[ J'_\mu(\pi) = \sum_s \mu(s) v_\pi(s) \]

Idea proposed in 2012.

Key lemma in the paper

\[ \nabla J'_\mu(\pi_\theta) \approx \sum_s \mu(s) \sum_a \pi_\theta(a|s) \nabla \pi_\theta(a|s) Q_\pi_\theta(s, a) \]

turns out to be wrong!

Still used as a heuristic justification of many algorithms!

Explicit formula for \( \nabla J'_\mu(\pi_\theta) \) can be obtained but much harder to use...
Outline

1 Policy Gradient Theorems
2 Monte Carlo Based Policy Gradient
3 Actor / Critic Principle
4 3 SOTA Algorithms
5 Average Return
6 References
PPO: Minorize-Majorization Algorithm

\[
J_{\mu_0}(\pi') \geq J_{\mu_0}(\pi) + \sum_t \gamma^t \mathbb{P}_\pi(S_t = s) \left( \sum_a (\pi'(s|a) - \pi(s|a)) A_\pi(s, a) \right) - \sum_t \gamma^t 2t \max_s \|\pi'(\cdot|s) - \pi(\cdot|s)\|_2^2 \max_{s,a} |A_\pi(s, a)|
\]

Ideal Minorize-Majorization Algorithm

- At step \( k \), find \( \theta_{k+1} \) maximizing

\[
J_{\mu_0}(\pi_{\theta} | \pi_{\theta_k}) = \sum_s \sum_t \gamma^t \mathbb{P}_{\pi_{\theta_k}}(S_t = s) \left( \sum_a (\pi_{\theta}(s|a) - \pi_{\theta_k}(s|a)) A_{\pi_{\theta_k}}(s, a) \right) - \sum_t \gamma^t 2t \max_s \|\pi_{\theta}(\cdot|s) - \pi_{\theta_k}(\cdot|s)\|_2^2 \max_{s,a} |A_{\pi_{\theta_k}}(s, a)|
\]

- By construction, \( J_{\mu_0}(\pi_{\theta_{k+1}}) \geq J_{\mu_0}(\pi_{\theta_k}) \)

- Sample efficient algorithm as the same trajectory can be (re)used in the optimization.
PPO: Optimization

\[ J_{\mu_0}(\pi_\theta) \geq J_{\mu_0}(\pi_{\theta_k}) + \sum_s \sum_t \gamma^t P_{\pi_{\theta_k}}(S_t = s) \left( \sum_a (\pi_\theta(s|a) - \pi_{\theta_k}(s|a)) A_{\pi_{\theta_k}}(s, a) \right) \]

\[- \sum_t \gamma^t 2t \max_s ||\pi_\theta(\cdot|s) - \pi_{\theta_k}(\cdot|s)||^2_1 \max_{s,a} |A_{\pi_{\theta_k}}(s, a)| \]

**Optimization**

- Gradient descent is possible.
- Gradient of the first term is

\[ \sum_s \sum_t \gamma^t P_{\pi}(S_t = s) \left( \sum_a \pi_\theta \nabla \pi_\theta(s|a) A_{\pi_{\theta_k}}(s, a) \right) \]

- Gradient of the second term more involved.

- Simpler (TRPO like) strategy: optimize

\[ \sum_s \sum_t \gamma^t P_{\pi_{\theta_k}}(S_t = s) \left( \sum_a (\pi_\theta(s|a) - \pi_{\theta_k}(s|a)) A_{\pi_{\theta_k}}(s, a) \right) \]

under \( \max_s ||\pi_\theta(\cdot|s) - \pi_{\theta_k}(\cdot|s)||^2_1 \leq \epsilon \) and reduce \( \epsilon \) there is no gain.
PPO: KL Relaxation

$$J_{\mu_0}(\pi_\theta) \geq J_{\mu_0}(\pi_{\theta_k}) + \sum_s \sum_t \gamma^t P_{\pi_{\theta_k}}(S_t = s) \left( \sum_a (\pi_\theta(s|a) - \pi_{\theta_k}(s|a)) A_{\pi_{\theta_k}}(s, a) \right)$$

$$- \frac{2\gamma R_{\text{max}}}{(1 - \gamma)^3} \max_s \text{KL}(\pi_{\theta_k}(.|s), \pi_\theta(.|s))$$

TRPO/PPO Optimization

- Replace the $\ell_1$ norm by a KL divergence.
- In practice, replace the max by an average and replace $\frac{2\gamma R_{\text{max}}}{(1 - \gamma)^3}$ by parameter $\beta$.
- PPO: Gradient descent of the relaxed goal.
- TRPO: Constrained optimization.

- Adaptive scheme to set $\beta$.
- Can be used with continuous action.
PPO: Clipped Objective

$$\sum_s \sum_t \gamma^t \mathbb{P}_{\pi_{\theta_k}} (S_t = s) \left( \sum_a \pi_{\theta_k} (s | a) \min \left( \frac{\pi_{\theta}(s | a)}{\pi_{\theta_k}(s, a)} A_{\pi_{\theta_k}} (s, a), \text{clip}(1 - \epsilon, \frac{\pi_{\theta}(s | a)}{\pi_{\theta_k}(s, a)}, 1 + \epsilon) A_{\pi_{\theta_k}} (s, a) \right) \right)$$

### Clipped Objective

- **Insight by (re)substracting** $$\sum_a \pi_{\theta_k} (s | a) A_{\theta_k} (s, a) = 0$$:
  
  $$\sum_a \min \left( (\pi_{\theta}(s | a) - \pi_{\theta_k} (s, a)) A_{\pi_{\theta_k}} (s, a), \text{clip}(-\epsilon, \pi_{\theta}(s | a) - \pi_{\theta_k} (s, a), \epsilon) A_{\pi_{\theta_k}} (s, a) \right)$$
  
  $$= \sum_a \text{clip}(-\epsilon \pi_{\theta_k} (s, a), \pi_{\theta}(s | a) - \pi_{\theta_k} (s, a), \epsilon \pi_{\theta_k} (s, a)) A_{\pi_{\theta_k}} (s, a)$$
  
  $$- \max \left( 0, - (\pi_{\theta}(s | a) - \pi_{\theta_k} (s, a)) A_{\pi_{\theta_k}} (s, a) - \epsilon \pi_{\theta_k} (s, a) | A_{\pi_{\theta_k}} (s, a) \right)$$

- **First term amount to replace** $$\pi_{\theta}$$ by a policy
  
  $$\tilde{\pi}_{\theta} (a | s) = \text{clip}(\pi_{\theta_k} (a | s)(1 - \epsilon), \pi_{\theta} (a | s), \pi_{\theta} (a | s)(1 + \epsilon)) + \eta_s \pi_{\theta_k} (a | s)$$

  where $$\eta$$ is so that $$\tilde{\pi}$$ is a probability for all $$s$$ and $$\|\tilde{\pi}_{\theta}(\cdot, s) - \pi_{\theta_k}(\cdot, s)\|_1 \leq \epsilon$$

- **Second term is a hinge loss type penalization of the policy** $$\pi_{\theta}$$ penalizing bad actions.
PPO: Stationary Objective

\[
\sum_s P_{\pi_{\theta_k}}(S_t = s) \left( \sum_a (\pi_{\theta}(s|a) - \pi_{\theta_k}(s|a)) A_{\pi_{\theta_k}}(s, a) \right) - \beta \max_s KL(\pi_{\theta_k}(\cdot|s), \pi_{\theta}(\cdot|s))
\]

\[
\sum_s P_{\pi_{\theta_k}}(S_t = s) \left( \sum_a \pi_{\theta_k}(s|a) \min \left( \frac{\pi_{\theta}(s|a)}{\pi_{\theta_k}(s, a)}, \text{clip}(1 - \epsilon, \frac{\pi_{\theta}(s|a)}{\pi_{\theta_k}(s, a)}, 1 + \epsilon) A_{\pi_{\theta_k}}(s, a) \right) \right)
\]

Stationary Objective

- Amount to replace $J_{\mu_0}(\pi)$ by $J_{\mu_{\pi}}(\pi)$
- Most common implementation of PPO...
- But no way to justify it mathematically!
- May explain the (possible) instabilities of PPO.
3 SOTA Algorithms

**DPG: Deterministic Policy Gradient**

\[ \pi_\theta(a|s) = 1_{a=h_\theta(s)} \quad \text{(deterministic policy)} \]

\[ J_{\mu_0}(\pi_\theta) = \sum_s \mu_0(s)v_{\pi_\theta}(s) \]

\[ \nabla J_{\mu_0}(\pi_\theta) = \sum_s \sum_t \gamma^t P_{\pi_\theta}(S_t = s) \nabla_a Q(S_t, h_\theta(S_t)) \]

---

**Deterministic Policy Gradient**

- Deterministic policy replaced by a randomized one centered on \( h_\theta(s) \) in the interactions!
- Critic trained with a TD variant of DQN.
- Gradient can be obtained by use a policy \( \pi_\theta = \mathcal{N}(h_\theta(s), \sigma^2 \text{Id}) \) and letting \( \sigma \) goes to 0.
- Off-Policy as claimed?
- Yes for the actor but no theoretical justification for the critic!
- In practice, the buffer contains only samples using a policy close to the current one...


SAC: A New Goal

\[ R_t \rightarrow R_t + \lambda \mathcal{H}(\pi(S_t)) \]

A Modified Reward

- Modification of the reward to favor high entropy policy:
  \[ R_t \rightarrow R_t + \lambda \mathcal{H}(\pi(S_t)) \]

- Goal:
  \[ J(\pi) = \sum_t (R_t + \lambda \mathcal{H}(\pi(S_t))) \]

- Soft value function implicitly defined as the fixed point of
  \[ \mathcal{T}^{\pi} Q_\pi(s, a) = r_\pi(s, a) + \sum_{s'} p(s'|s, a) V_\pi(s') \]

where

\[ V_\pi(s, a) = \sum_a \pi(a|s) (Q_\pi(s, a) - \log \pi(a|s)) \]
SAC: Policy Improvement and Optimal Policy

\[ R_t \rightarrow R_t + \lambda \mathcal{H}(\pi(S_t)) \]

### A Modified Policy Improvement Lemma

- **Policy improvement rule:**

  \[
  \pi^+(\cdot|s) = \arg\max_{\pi(\cdot|s)} \sum_a \pi(a|s) (q(s,a) - \lambda \log(\pi(a|s)))
  \]

  \[
  \pi^+(a|s) \propto \exp\left(-\frac{1}{\lambda}q(s,a)\right)
  \]

  implies \( G_{\pi^+}(s,a) \geq G_{\pi}(s,a) \).

- **At convergence,** \( J(\pi^*) \) is optimal!

- **Convergence in the finite setting.**
SAC: Parametrization

\[ \pi \sim \pi_\theta \quad \text{and} \quad Q(s, a) \sim Q_w \]

### SAC Choices

- **Fitted TD learning for** \( Q \):
  \[
  w \simeq \text{argmin} \sum_{(S,A,R,S') \in B} \left( R + \mathbb{E}_{\pi_\theta} \left[ \gamma Q_w(S', a) - \lambda \log \pi_\theta(a|S') \right] - Q_w(S, A) \right)^2
  \]
  where the trajectory pieces are samples from a replay buffer and \( w \) is a slowdown version of \( w \) (two scale algorithm).

- **Online version rather than batch**... 

- **Fitted KL for** \( \pi \):
  \[
  \theta \simeq \text{argmin} \sum_{(S,A,R,S') \in B} \text{KL}(\pi_\theta(\cdot|S)\| \exp -\lambda Q_w(S, \cdot)/Z_w(S))
  \]
  \[
  \simeq \sum_{(S,A,R,S') \in B} \mathbb{E}_{\pi_\theta} \left[ \frac{1}{\lambda} \log \pi_\theta(a|S) - Q_\theta(a|S) \right]
  \]
Outline

1. Policy Gradient Theorems
2. Monte Carlo Based Policy Gradient
3. Actor / Critic Principle
4. 3 SOTA Algorithms
5. Average Return
6. References
Continuing Tasks and Average Return

• Most natural performance measure:

\[ J(\pi) = r(\pi) = \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_{\pi}[R_t | S_0] \]

\[ = \sum_{s} \left( \lim_{T \to \infty} \sum_{t=1}^{T} \mathbb{P}_{\pi}(S_t = s | S_0) \right) \sum_{a} \pi(a|s) \sum_{s',r} p(s', r | s, a) r \]

\[ \mu(s) \]

• \( \mu \) if it exists is such that

\[ \sum_{s} \sum_{a} \mu(s) \pi(a|s)p(s'|s, a) = \mu(s') \]

• Gradient of \( J(\pi_\theta) \):

\[ \nabla J(\pi_\theta) = \sum_{s} \mu(s) \sum_{a} \pi_\theta(a|s) \nabla \log \pi_{\pi_\theta}(a|s) q_{\pi_\theta}(s, a) \]

• Beware \( q_{\pi_\theta} \) are the relative Q-value functions and not the classical one.
Average Return and Relative Value Functions

\[ r(\pi) = \sum_s \mu(s) \sum_a \pi(a|s) \sum_{s', r} p(s', r|s, a)r \]

\[ G_t = \sum_{t' \geq t} R_{t'} - r(\pi) \]

\[ V_\pi(s) = \mathbb{E}_\pi[G_t|S_t = s] \quad \text{and} \quad Q_\pi(s) = \mathbb{E}_\pi[G_t|S_t = s] \]

- Numerical algorithm to estimate those relative value functions.
- Leads to another family of Policy Gradient algorithm.
Outline

1. Policy Gradient Theorems
2. Monte Carlo Based Policy Gradient
3. Actor / Critic Principle
4. 3 SOTA Algorithms
5. Average Return
6. References
References

R. Sutton and A. Barto.  
*Reinforcement Learning, an Introduction (2nd ed.)*  
MIT Press, 2018

O. Sigaud and O. Buffet.  
*Markov Decision Processes in Artificial Intelligence.*  
Wiley, 2010

M. Puterman.  
Wiley, 2005

D. Bertsekas and J. Tsitsiklis.  
*Neuro-Dynamic Programming.*  
Athena Scientific, 1996

W. Powell.  
Wiley, 2022

S. Meyn.  
*Control Systems and Reinforcement Learning.*  
Cambridge University Press, 2022

V. Borkar.  
*Stochastic Approximation: A Dynamical Systems Viewpoint.*  
Springer, 2008

T. Lattimore and Cs. Szepesvári.  
*Bandit Algorithms.*  
Cambridge University Press, 2020
Licence and Contributors

Creative Commons Attribution-ShareAlike (CC BY-SA 4.0)

- You are free to:
  - **Share**: copy and redistribute the material in any medium or format
  - **Adapt**: remix, transform, and build upon the material for any purpose, even commercially.

- Under the following terms:
  - **Attribution**: You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
  - **ShareAlike**: If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.
  - **No additional restrictions**: You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Contributors

- Main contributor: E. Le Pennec