Theory and Methods for Reinforcement Learning

Prof. Volkan Cevher
volkan.cevher@epfl.ch

Lecture 8: Policy Gradient Methods II

Laboratory for Information and Inference Systems (LIONS)
École Polytechnique Fédérale de Lausanne (EPFL)

EE-618 (Spring 2020)

License Information for Reinforcement Learning Slides

- This lecture
  1. Trust Region Policy Optimization
  2. Proximal Policy Optimization
- Next lecture
  1. Actor-Critic Methods for Deep RL

Recommended Reading

- Schulman, John and Levine, Sergey and Abbeel, Pieter and Jordan, Michael and Moritz, Philipp. *Trust region policy optimization.*, International Conference on Machine Learning, 2015.
- Schulman, John and Wolski, Filip and Dhariwal, Prafulla and Radford, Alec and Klimov, Oleg. *Proximal policy optimization algorithms*, arXiv preprint, 2017.

Policy Gradient Method

- Policy gradient methods (PGM):
  - Parametrize a stochastic policy using parameter vector θ
  - Define a performance measure $J(\theta) = \mathbb{E}_{s_0} v_{\pi_\theta}(s_0)$
  - Maximize J over θ using Stochastic Gradient Descent:

$$\theta_{t+1} = \theta + \alpha \widehat{\nabla J}(\theta_t)$$

- Goal of this lecture: Present two state-of-the-art practical algorithms based on PGM.

Trust Region Policy Optimization

- Trust Region Policy Optimization (TRPO) [3]:
  - Introduces a surrogate objective performance measure.
  - Designs a theoretical update scheme which iteratively updates a policy in a way that guarantees **monotone improvement**, i.e., ensures $J(\pi_{t+1}) \geq J(\pi_t)$.
  - Approximates this theoretical scheme using optimization tools.

TRPO: Preliminaries

Lemma

Given any two policies π, $\tilde{\pi}$,

$$J(\tilde{\pi}) = J(\pi) + \mathbb{E}_{\tilde{\pi}}\left[\sum_{t=0}^{\infty} \gamma^t a_\pi(S_t, A_t)\right] \tag{1}$$

where $a_\pi(s,a) = q_\pi(s,a) - v_\pi(s)$ is the advantage function.

Proof:
First note that $a_\pi(s,a) = \mathbb{E}_{s' \sim p(s'|s,a)}[r(s) + \gamma v_\pi(s') - v_\pi(s)]$. Therefore,

$$\begin{aligned}
\mathbb{E}_{\tilde{\pi}}\left[\sum_{t=0}^{\infty} \gamma^t a_\pi(S_t, A_t)\right] &= \mathbb{E}_{\tilde{\pi}}\left[\sum_{t=0}^{\infty} \gamma^t (r(S_t) + \gamma v_\pi(S_{t+1}) - v_\pi(S_t))\right] \\
&= \mathbb{E}_{\tilde{\pi}}\left[-v_\pi(s_0) + \sum_{t=0}^{\infty} \gamma^t r(S_t)\right] \\
&= -\mathbb{E}_{s_0}[v_\pi(s_0)] + \mathbb{E}_{\tilde{\pi}}\left[\sum_{t=0}^{\infty} \gamma^t r(S_t)\right] \\
&= -J(\pi) + J(\tilde{\pi})
\end{aligned}$$

TRPO: Preliminaries

Equation (1) can be rewritten with a sum over states instead of time steps:

$$
\begin{aligned}
J(\tilde{\pi}) &= J(\pi) + \sum_{t=0}^{\infty} \sum_{s} P(S_t = s|\tilde{\pi}) \sum_{a} \tilde{\pi}(a|s)\gamma^t a_\pi(a, s) \\
&= J(\pi) + \sum_{s} \sum_{t=0}^{\infty} \gamma^t P(S_t = s|\tilde{\pi}) \sum_{a} \tilde{\pi}(a|s) a_\pi(a, s) \\
&= J(\pi) + \sum_{s} \rho_{\tilde{\pi}}(s) \sum_{a} \tilde{\pi}(a|s) a_\pi(a, s) \qquad (2)
\end{aligned}
$$

where $\rho_{\tilde{\pi}}(s) = \sum_{t=0}^{\infty} \gamma^t P(S_t = s|\tilde{\pi})$ is the unnormalized discounted visitation frequency.

TRPO: Preliminaries

$$J(\tilde{\pi}) = J(\pi) + \sum_s \rho_{\tilde{\pi}}(s) \sum_a \tilde{\pi}(a|s) a_\pi(a,s) \tag{3}$$

• Equation (3) gives an alternative way of optimizing the performance J, by maximizing the right hand side for any fixed policy π.

• However, the complex dependency of $\rho_{\tilde{\pi}}$ on $\tilde{\pi}$ makes it difficult to optimize directly. Instead, let's introduced the following surrogate to J:

$$L_\pi(\tilde{\pi}) = J(\pi) + \sum_s \rho_\pi(s) \sum_a \tilde{\pi}(a|s) a_\pi(a,s) \tag{4}$$

i.e. $\rho_{\tilde{\pi}}$ is simply replaced by ρ_π.

• It is easy to check that for any differentiable parametrized policy π_θ and $\pi_{\theta_{old}}$,

$$L_{\pi_{\theta_{old}}}(\pi_{\theta_{old}}) = J(\pi_{\theta_{old}}) \tag{5}$$

$$\nabla_\theta L_{\pi_{\theta_{old}}}(\pi_\theta)\Big|_{\theta=\theta_{old}} = \nabla_\theta J(\pi_\theta)|_{\theta=\theta_{old}} \,. \tag{6}$$

$\Rightarrow$ a sufficiently small step $\pi_{\theta_{old}} \to \tilde{\pi}$ that improves $L_{\pi_{\theta_{old}}}$ will also improve J!

Monotonic Improvement Guarantee for General Stochastic Policies

- Question: How big of a step to take?

Theorem

For any two stochastic policies $\pi, \tilde{\pi}$*, we have*

$$J(\tilde{\pi}) \;\geq\; L_\pi(\tilde{\pi}) - C D_{KL}^{max}(\pi, \tilde{\pi}) \tag{7}$$

where $D_{KL}^{max}(\pi, \tilde{\pi}) = \max_s D_{KL}\left(\pi(.|s)||\tilde{\pi}(.|s)\right)$*,* $C = \frac{4\epsilon\gamma}{(1-\gamma)^2}$ *and* $\epsilon = \max_{s,a} |A_\pi(s,a)|$*.*

Proof: on the board, or see [2, 3].

- Thanks to (5), we see that this upper bound becomes tight for $\pi = \tilde{\pi}$.

Monotonic Improvement Guarantee for General Stochastic Policies

$$
\begin{aligned}
J(\tilde{\pi}) &\geq L_{\pi}(\tilde{\pi}) - CD_{KL}^{max}(\pi, \tilde{\pi}) \\
J(\pi_{\theta_{old}}) &= L_{\pi_{\theta_{old}}}(\pi_{\theta_{old}})
\end{aligned}
$$

- For any fixed policy π, maximizing the RHS of (7) necessary leads to a new policy with better performance than with π, i.e.,

$$
\tilde{\pi}^* = \arg\max_{\tilde{\pi}} L_{\pi}(\tilde{\pi}) - CD_{KL}^{max}(\pi, \tilde{\pi}) \Rightarrow J(\tilde{\pi}^*) \geq J(\pi)
$$

- To see this, let $M_i(\pi) = L_{\pi_i}(\pi) - CD_{KL}^{max}(\pi_i, \pi)$. Then

$$
\begin{aligned}
J(\pi_{i+1}) &\geq M_i(\pi_{i+1}) \text{ by } (7) \\
J(\pi_i) &= M_i(\pi_i) \text{ by } (5)\text{, therefore,} \\
J(\pi_{i+1}) - J(\pi_i) &\geq M_i(\pi_{i+1}) - M_i(\pi_i).
\end{aligned}
$$

Thus, by maximizing M_i at each iteration, we guarantee that the true objective J is non-decreasing.

Monotonic Improvement Guarantee for General Stochastic Policies

Algorithm 1 Policy iteration algorithm guaranteeing non-decreasing expected return

1: Initialize π_0
2: **for** i=0,1,2,... until convergence **do**
3: Compute all advantage values $A_{\pi_i}(s,a)$.
4: Solve

$$\pi_{i+1} = \arg\max_{\tilde{\pi}} L_{\pi_i}(\tilde{\pi}) - CD_{KL}^{max}(\pi_i, \tilde{\pi}), \quad (8)$$

where $C = \frac{4\epsilon\gamma}{(1-\gamma)^2}$

5: **end for**

• Algorithm 1 is guaranteed to generate a monotonically improving sequence of policies $J(\pi_0) \leq J(\pi_1) \leq J(\pi_2) \leq \ldots$.

Issues with Algorithm 1

- The theoretical constant turns out to be much too large in practice, leading to very small updates, and is difficult to choose manually.

- Computation of L_π requires knowledge of the advantage function A_π.

- We must efficiently solve the optimization problem (8) for updating the policy.

Replace Penalization by a Hard Trust Region Constraint

• Replace the penalization by a hard constraint on the KL divergence between the old and the new policy, i.e., a trust region constraint:

$$\begin{aligned} &\max_{\pi} L_{\pi_{old}}(\pi) \\ &\text{subject to } D_{KL}^{max}(\pi_{old}, \pi) \leq \delta, \end{aligned} \tag{9}$$

The upper bound δ on the maximum KL divergence turns out to be much less problem dependent, and easier to tune.

• But: The number of constraint is equal to $|\mathcal{S}|$!!

• Replace D_{KL}^{max} by the average KL divergence over all states:

$$\begin{aligned} &\max_{\pi} L_{\pi_{old}}(\pi) \\ &\text{subject to } \bar{D}_{KL}^{\rho_{\pi_{old}}}(\pi_{old}, \pi) \leq \delta \end{aligned} \tag{10}$$

where $\bar{D}_{KL}^{\rho_\pi}(\pi_1, \pi_2) = \mathbb{E}_{s \sim \rho_\pi}\left[D_{KL}(\pi_1(.|s), \pi_2(.|s))\right]$

Recasting Problem (10)

- Expand $L_{\pi_{old}}(\pi)$ in problem (10):

$$\begin{aligned} &\max_{\pi} \sum_{s} \rho_{\pi_{old}}(s) \sum_{a} \pi(a|s) a_{\pi_{old}}(a,s) \\ &\text{subject to } \bar{D}_{KL}^{\rho_{\pi_{old}}}(\pi_{old}, \pi) \leq \delta \end{aligned} \tag{11}$$

- We now make the following rewritings:
  - $\sum_s \rho_{\pi_{old}}[\ldots] \longrightarrow \frac{1}{1-\gamma}\mathbb{E}_{s\sim\rho_{\pi_{old}}}[\ldots]$
  - $a_{\pi_{old}} \longrightarrow q_{\pi_{old}}$ (only changes the objective by a constant)
  - $\sum_a \pi(a|s) a_{\pi_{old}}(s,a) = \mathbb{E}_{a\sim q}\left[\frac{\pi(a|s)}{q(a|s)} a_{\pi_{old}}(s,a)\right]$ $\forall s \in \mathcal{S}$ (important sampling)

- Problem (10) is then exactly equivalent to:

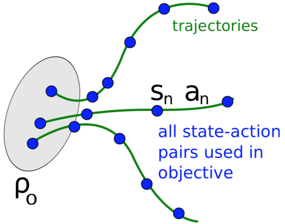
$$\begin{aligned} &\max_{\pi} \mathbb{E}_{s\sim\rho_{\pi_{old}}, a\sim\pi_{old}}\left[\frac{\pi(a|s)}{\pi_{old}(a|s)} Q_{\pi_{old}}(s,a)\right] \\ &\text{subject to } \mathbb{E}_{s\sim\rho_{\pi_{old}}}[D_{KL}(\pi_{old}(.|s), \pi(.|s))] \leq \delta \end{aligned} \tag{12}$$

Sampled-Based Estimation of The Objective and Constraint I

- Approximate expectations by sample averages, and the Q values by empirical average:

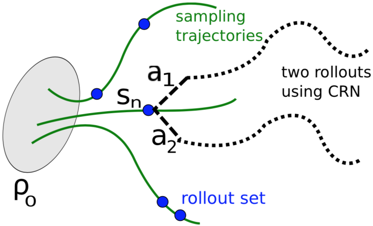
$$
\begin{aligned}
&\max_{\pi} \hat{L}_{\pi_{old}}(\pi) := \sum_{i=1}^{N} \frac{\pi(a_i|s_i)}{\pi_{old}(a_i|s_i)} \hat{Q}_{\pi_{old}}(s_i, a_i) \\
&\text{subject to} \sum_{i=1}^{N} [D_{KL}(\pi_{old}(.|s_i), \pi(.|s_i))] \leq \delta
\end{aligned}
\tag{13}
$$

- Single path scheme: Standard Monte-Carlo estimation



Sampled-Based Estimation of The Objective and Constraint II

• Vine scheme: Evaluate the Q values independently for state-action pairs (s, a) encountered during various trajectories.



• Advantage:

- Provides samples with much lower variance

• Drawbacks:

- Requires more calls to the simulator
- Requires to generate multiple trajectories from prescribed states

Natural Policy Gradient I

$$\max_{\theta} \mathbb{E}_{s\sim\rho_{\pi_{\theta_{old}}}, a\sim\pi_{\theta_{old}}}\left[\frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)} Q_{\pi_{\theta_{old}}}(s,a)\right]$$
$$\text{subject to } \mathbb{E}_{s\sim\rho_{\pi_{\theta_{old}}}}\left[D_{KL}(\pi_{\theta_{old}}(.|s), \pi_\theta(.|s))\right] \le \delta \tag{14}$$

- NPG approximates problem (14) by using a linear approximation to the objective and a quadratic approximation to the constraint around parameters θ_{old}, i.e.,

$$\max_{\theta} g \cdot (\theta - \theta_{old})$$
$$\text{subject to } \frac{1}{2}(\theta_{old} - \theta)^T \bar{A}_{\theta_{old}}(\theta_{old} - \theta) \le \delta,$$

where $g \simeq \nabla_\theta \hat{L}_{\pi_{\theta_{old}}}(\theta)\Big|_{\theta=\theta_{old}}$ and

$$(\bar{A}_{\theta_{old}})_{ij} = \frac{\partial^2}{\partial\theta_i \partial\theta_j} \bar{D}_{KL}^{\rho_{\pi_{old}}}(\pi_{\theta_{old}}, \pi_\theta)\Bigg|_{\theta=\theta_{old}}$$
$$\simeq \frac{1}{N}\sum_{i=1}^{N} \frac{\partial^2}{\partial\theta_i \partial\theta_j} D_{KL}(\pi_{\theta_{old}}(\cdot|s_i), \pi_\theta(\cdot|s_i))\Bigg|_{\theta=\theta_{old}}$$

is the average Fisher Information Matrix (FIM).

Natural Policy Gradient II

$$\max_{\theta} g \cdot (\theta - \theta_{old})$$
$$\text{subject to } \frac{1}{2}(\theta_{old} - \theta)^T \bar{A}_{\theta_{old}} (\theta_{old} - \theta) \leq \delta, \tag{15}$$

- Problem (15) is a quadratic equation and can be solved analytically:

$$\theta^* = \theta_{old} + \sqrt{\frac{2\delta}{g^T \bar{A}_{\theta_{old}}^{-1} g}} \bar{A}_{\theta_{old}}^{-1} g$$

- Limitations of NPG:
  - Finding the inverse of $\bar{A}_{\theta_{old}}$ is expensive
  - Given the search direction $\bar{A}_{\theta_{old}}^{-1} g$, the step size $\sqrt{\frac{2\delta}{g^T \bar{A}_{\theta_{old}}^{-1} g}}$ may not be optimal.

Practical Algorithm: TRPO

$$\begin{aligned} &\max_{\theta} g \cdot (\theta - \theta_{old}) \\ &\text{subject to } \frac{1}{2}(\theta_{old} - \theta)^T \bar{A}_{\theta_{old}} (\theta_{old} - \theta) \leq \delta, \end{aligned} \tag{16}$$

- In order to solve problem (16), we repeat the following steps until convergence:

1. Compute a search direction $s \simeq \bar{A}_{\theta_{old}}^{-1} g$ using conjugate gradient algorithm.
2. Perform a line search in this direction, starting from proposed step $\sqrt{\frac{2\delta}{s^T \bar{A}_{\theta_{old}} s}}$.
3. Update the policy parameters $\theta \leftarrow \theta + \beta s$, where, s, β are the resulting direction and step-size computed at steps 1,2 respectively.

TRPO Algorithm

TRPO [1]

Algorithm parameter: Initial policy parameters θ_0, KL divergence constraint parameter δ

for k = 0,1,2,. . . **do**

Collect set of trajectories on policy π_{θ_k}
Estimate Q values using *single path* or *vine* sampling scheme
Use CG algorithm to obtain $s_k \simeq \bar{A}_{\theta_k}^{-1} g_k$
Estimate proposed step $\Delta_k \simeq \sqrt{\frac{2\delta}{s_k^T \bar{A}_{\theta_k}^{-1} s_k}} s_k$.
Perform backtracking line search with exponential decay, starting from Δ_k to obtain step-size β_k
Update the policy parameters:

$$\theta_{k+1} \leftarrow \theta_k + \beta_k s_k$$

end for

Drawbacks of TRPO

- Relatively complicated
- Not compatible with architecture involving noise (such as dropout) or parameter sharing
- Less sample efficient than methods trained using first-order optimizers such as Adam

- Question: Following similar ideas, can we design a simpler algorithm at least as performant as TRPO?

Proximal Policy Optimization (PPO) with Adaptive KL Penalty [4]

- Going back to penalized problem:

$$\max_{\theta} L_{\pi_{\theta_k}}(\pi_\theta) - C\bar{D}_{KL}^{\rho_{\pi_{\theta_k}}}(\pi_{\theta_k}, \pi_\theta)$$

PPO with Adaptive KL Penalty [1]

Algorithm parameter: Initial policy parameters θ_0, initial KL penalty β_0, target KL-divergence δ.

for k = 0,1,2,. . . **do**
  Collect set of trajectories on policy π_{θ_k}
  Estimate Q values using *single path* or *vine* sampling scheme
  Compute policy upate

$$\theta_{k+1} = \arg\max_{\theta} L_{\pi_{\theta_k}}(\pi_\theta) - \beta_k \bar{D}_{KL}^{\rho_{\pi_{\theta_k}}}(\pi_{\theta_k}, \pi_\theta)$$

  using Adam.
  if $\bar{D}_{KL}^{\rho_{\pi_{\theta_k}}}(\pi_{\theta_k}, \pi_{\theta_{k+1}}) \geq 1.5\delta$ **then**
    $\beta_{k+1} \leftarrow 2\beta_k$
  else if $\bar{D}_{KL}^{\rho_{\pi_{\theta_k}}}(\pi_{\theta_k}, \pi_{\theta_{k+1}}) \leq \frac{\delta}{1.5}$ **then**
    $\beta_{k+1} \leftarrow \frac{\beta_k}{2}$
  end if
end for

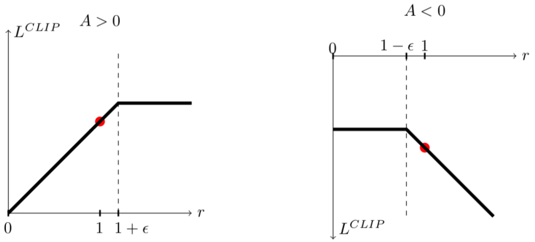
Proximal Policy Optimization (PPO) [4]

- PPO replaces the TRPO objective of equation (12)

$$L_{\pi_{old}}(\pi) = \mathbb{E}_{s\sim\rho_{\pi_{old}}, a\sim\pi_{old}}\left[\frac{\pi(a|s)}{\pi_{old}(a|s)} a_{\pi_{old}}(a,s)\right] \equiv L^{CPI}(\pi)$$

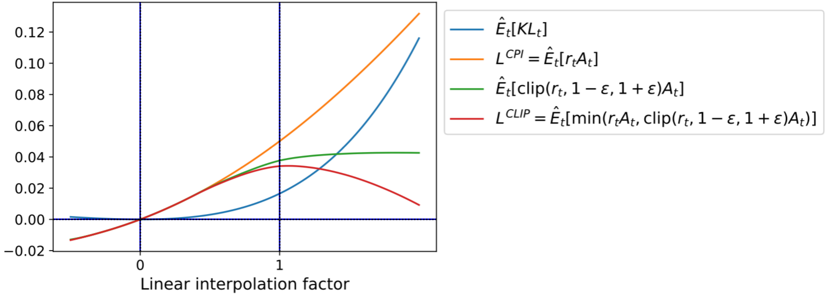
with the following clipped version (ϵ usually set to 0.2):

$$L^{CLIP}(\pi) = \mathbb{E}_{s\sim\rho_{\pi_{old}}, a\sim\pi_{old}}\Bigg[\min\left(\frac{\pi(a|s)}{\pi_{old}(a|s)} a_{\pi_{old}}(a,s),\right.$$
$$\left.\text{clip}\left(\frac{\pi(a|s)}{\pi_{old}(a|s)}, 1-\epsilon, 1+\epsilon\right) a_{\pi_{old}}(a,s)\right)\Bigg]$$



PPO Objective Function

- First policy update on the Hopper-v1 problem:



- PPO penalizes large deviation from the current policy directly inside the objective function

PPO Algorithm

PPO with Clipped Objective [1]

Algorithm parameter: Initial policy parameters θ_0, clipping threshold ϵ.

for k = 0,1,2,... **do**

Collect set of trajectories on policy π_{θ_k}

Estimate Q values using *single path* or *vine* sampling scheme

Compute policy upate

$$\theta_{k+1} = \arg\max_{\theta} L^{CLIP}_{\pi_{\theta_k}}(\pi_\theta)$$

using Adam.

end for

References

[1] Joshua Achiam.
Advanced policy gradient methods, 2017.

[2] Sham Kakade and John Langford.
Approximately optimal approximate reinforcement learning.
In *ICML*, volume 2, pages 267–274, 2002.

[3] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz.
Trust region policy optimization.
In *International Conference on Machine Learning*, pages 1889–1897, 2015.

[4] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov.
Proximal policy optimization algorithms.
arXiv preprint arXiv:1707.06347, 2017.

Computing the Fisher-Vector Product

• Applying conjugate gradient algorithm in step 1 requires to perform matrix-vector multiplication with the FIM. We show here how to make this computation efficiently.
• Suppose that the parametrized policy maps from the state s to distribution parameter vector $\mu_\theta(s)$, which parametrizes the distribution $\pi(a|s)$.
Let $A_{\theta_{old}}(s) = \frac{\partial^2}{\partial\theta_i \partial\theta_j} D_{KL}(\pi_{\theta_{old}}(\cdot|s), \pi_\theta(\cdot|s))\Big|_{\theta=\theta_{old}}$ be the FIM of the policy at state s. Then, $A_{\theta_{old}}(s)$ can be written as

$$A_{\theta_{old}}(s) = J^T M J$$

where $J := \frac{\partial \mu_\theta(s)}{\partial\theta}$ is the Jacobian of $\mu_\theta(s)$, and
$M = \mathbb{E}\left[\left(\frac{\partial \log \pi_\theta(\cdot|s)}{\partial \mu_\theta(s)}\right)\left(\frac{\partial \log \pi_\theta(\cdot|s)}{\partial \mu_\theta(s)}\right)^T \middle| \theta = \theta_{old}\right]$ is the FIM of the distribution $\pi_\theta(\cdot|s)$ in terms of the parameter μ_θ (as opposed to the parameter θ), which has a simple form for most parametrized distributions of interest.

The Fisher-vector product can now be written as a function $y \to J^T M T y$.
Multiplication by J^T and J can be performed by most automatic differentiation and neural network packages (multiplication by J^T is the well-known backprop operation), and the multiplication by M can be derived for the distribution of interest.