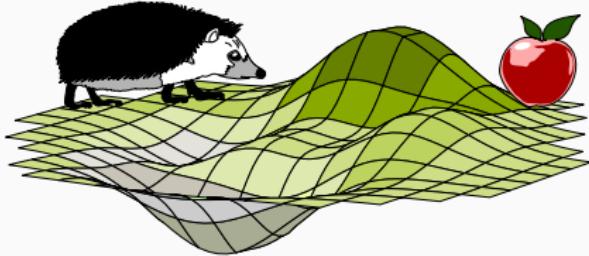


REINFORCEMENT LEARNING

POLICY GRADIENT

Pavel Osinenko





Consider an environment

$$X_{n+1} \sim P_X(x_{n+1} | x_n, u_n)$$

$$U_n \sim P_U^{\vartheta}(u_n | x_n)$$

along with an ∞ -horizon optimal control problem

$$\max_{\vartheta} J(x_0 | \vartheta) = E \left[\sum_{k=0}^{\infty} \gamma^k p(X_k, U_k) | X_0 = x_0 \right]$$

We seek to optimize $J(\cdot | \vartheta)$ via
the distribution parameters ϑ



(cont.)

Let's introduce the following notation:

$$Z := (X, U)$$

(concatenation of state-action pairs)

Thus, the environment now reads:

$$Z_{k+1} \sim P_Z^{\pi}(Z_{k+1} | Z_k)$$

So, the process $\{Z_k\}_k$ is interpreted as
state - action trajectory

Policy gradient



(cont.)

Introduce the following random variable:

$$\bar{Z}_k^N := \{Z_k, Z_{k+1}, \dots, Z_{k+N}\}$$

Consider the accumulated N-horizon reward:

$$\bar{R}_N(\bar{Z}_0^N) := \sum_{k=0}^N \gamma^k p(Z_k)$$

where γ is the discount factor. Remember that the transition probability of Z depends on ϑ . Let us now concentrate on optimization of the finite-horizon objective

$$J_N(x_0 | \vartheta) = \mathbb{E} \left[\bar{R}_N(\bar{Z}_0^N) \mid X_0 = x_0 \right]$$

Policy gradient



(cont.)

Let's unwrap the objective:

$$\begin{aligned} J_N(x_0 | \varphi) &= \mathbb{E} \left[\bar{R}_N(\bar{z}_0^N) \mid X_0 = x_0 \right] \\ &= \int_{\mathcal{X}^N \times \mathcal{U}^N} \bar{R}_N(\bar{z}_0^N) \mathbb{P} \left[d\bar{z}_0^N \mid X_0 = x_0 \right], \end{aligned}$$

$$\text{where } \bar{z}_0^N = \{z_0, \dots, z_N\}$$

Unwrapping the probability measure yields for $J_N(x_0 | \varphi)$:

$$\int_{\mathcal{X}^N \times \mathcal{U}^N} \bar{R}_N(\bar{z}_0^N) P_X(x_0) \cdot \prod_{k=0}^N \left(P_X(x_{k+1} | x_k, u_k) \cdot P_U^\varphi(u_k | x_k) \right) d\bar{z}_0^N$$

Policy gradient



(cont.)

One of the most straightforward ways to optimize the objective J is to apply gradient ascent along ϑ to it:

$$\vartheta^{i+1} := \vartheta^i + \alpha \nabla_{\vartheta} J_N(x_0 | \vartheta^i),$$

where i is the gradient ascent iteration index and α is the learning rate.

Now, we need to figure out the gradient. We will use the log-likelihood trick:

$$\nabla_{\vartheta} P_Z^{\vartheta}(z) = P_Z^{\vartheta}(z) \nabla_{\vartheta} \ln P_Z^{\vartheta}(z)$$

Verify!

Policy gradient



(cont.)

Applying the trick yields:

$$\nabla_{\theta} \left(\int_{\mathcal{X}^N \times \mathcal{U}^N} \bar{R}_N(\bar{\pi}^N) P_X(x_0) \cdot \prod_{k=0}^N \left(P_X(x_{k+1} | x_k, u_k) \cdot P_U^\theta(u_k | x_k) \right) d\bar{\pi}^N \right) =$$

$$\int_{\mathcal{X}^N \times \mathcal{U}^N} \bar{R}_N(\bar{\pi}^N) P_X(x_0) \cdot \prod_{k=0}^N \left(P_X(x_{k+1} | x_k, u_k) \cdot P_U^\theta(u_k | x_k) \right) \cdot \nabla_{\theta} \left(\ln P_X(x_0) \cdot \prod_{k=0}^N \left(P_X(x_{k+1} | x_k, u_k) \cdot P_U^\theta(u_k | x_k) \right) \right) d\bar{\pi}^N$$

Policy gradient



(cont.)

Observe the following, & h:

$$\mathbb{E}[h(x)] = \int_{\mathcal{X}} P(x) h(x) dx = \int_{\mathcal{X}} h(x) P(dx)$$

So,

$$\int_{\mathcal{X}^N \times \mathcal{U}^N} P_X(x_0) \cdot \prod_{k=0}^N \left(P_X(x_{k+1} | x_k, u_k) \cdot P_U^\pi(u_k | x_k) \right) \bar{R}_N(\bar{\Sigma}_0^N) \cdot \nabla_\theta \left(\ln P_X(x_0) \cdot \prod_{k=0}^N \left(P_X(x_{k+1} | x_k, u_k) \cdot P_U^\pi(u_k | x_k) \right) \right) d\bar{\Sigma}_0^N$$

(also using the log-of-product)

$$= \mathbb{E} \left[\bar{R}_N(\bar{\Sigma}_0^N) \sum_{k=0}^N \nabla_\theta \ln P_U^\pi(u_k | x_k) \mid X_0 = x_0 \right] \text{ Verify!}$$

Policy gradient



(cont.)

Plugging this into the gradient ascent gives a policy algorithm called REINFORCE:

$$\begin{aligned}\vartheta^{i+1} &= \vartheta^i + \alpha \mathbb{E} \left[\bar{R}_N(\bar{Z}_0) \sum_{k=0}^N \nabla_\vartheta \ln P_{\vartheta^i}^\pi(U_k | X_k) \mid X_0 = x_0 \right] \\ &= \vartheta^i + \alpha \mathbb{E} \left[\sum_{k=0}^N \nabla_\vartheta \ln P_{\vartheta^i}^\pi(U_k | X_k) \cdot \sum_{j=0}^N \gamma^j p(X_j, U_j) \mid X_0 = x_0 \right]\end{aligned}$$

In practice, \mathbb{E} is substituted for a sample mean and N is either fixed or the episodes are run until success or failure



As with usual gradient descent/ascent methods, numerous modifications are possible.

Say, in the **natural policy gradient** (in contrast to the **vanilla**, the „ordinary“ one), one may control the step size „ $\Delta \theta_k^i$ “ (in quotes because we are talking about distribution parameters).

This can be done, e.g., via the **Kullback - Leibler divergence** (omitting the $1x$ condition for brevity) :

$$d_{KL}(P_J^\sigma \| P_J^{\sigma + \Delta \sigma}) \triangleq \int \limits_{\mathcal{S}} P_J^\sigma(u) \ln \frac{P_J^\sigma(u)}{P_J^{\sigma + \Delta \sigma}(u)} du$$

Policy gradient



(cont.)

So, the algorithm would read, for instance:

$$\Delta \theta^* := \arg \max_{\Delta \theta} J(x | \theta_0 + \Delta \theta)$$

$$\text{s.t. } d_{KL}(P_J^{\theta_0} \| P_J^{\theta_0 + \Delta \theta}) \leq \epsilon ,$$

where ϵ is a tuning parameter to control the gradient ascent step size

(we omitted all indices for brevity except for θ_0 which indicates the current parameters)

Policy gradient



(cont.)

We can rewrite the above optimization problem using a Lagrange multiplier:

$$\Delta\vartheta^* := \arg \max_{\Delta\vartheta} \left\{ J(x|\vartheta_0 + \Delta\vartheta) - \lambda \left(d_{KL}(P_{\pi}^{\vartheta_0} \| P_{\pi}^{\vartheta_0 + \Delta\vartheta}) - \varepsilon \right) \right\}$$

Let's denote the Lagrangian as:

$$\mathcal{L}(\Delta\vartheta, \lambda) = J(x|\vartheta_0 + \Delta\vartheta) - \lambda \left(d_{KL}(P_{\pi}^{\vartheta_0} \| P_{\pi}^{\vartheta_0 + \Delta\vartheta}) - \varepsilon \right),$$

in other words, the penalized objective.

The necessary optimality conditions read:

$$\nabla_{\Delta\vartheta} \mathcal{L} = 0$$

$$\nabla_{\lambda} \mathcal{L} = 0$$



(cont.)

So, we need to find the respective gradients.

Let's use the Taylor expansion :

- for the objective :

$$J(x|\theta_0 + \Delta\theta) = J(x|\theta_0) + \nabla_{\theta} J(x|\theta_0) \Delta\theta + \mathcal{O}(\|\Delta\theta\|^2)$$

- for the KL-divergence :

$$d_{KL}(P_{\pi}^{\theta_0} \| P_{\pi}^{\theta_0 + \Delta\theta}) = 0 + \langle \text{1st order} \rangle + \langle \text{2nd order} \rangle + \mathcal{O}(\|\Delta\theta\|^3)$$

Policy gradient



(cont.)

Let's first work out the KL-divergence.

Recall the formula and expand it as:

$$\int \mathcal{P}_J^{\theta}(u) \ln \frac{\mathcal{P}_J^{\theta}(u)}{\mathcal{P}_J^{\theta} + \Delta\theta(u)} du = \int \mathcal{P}_J^{\theta}(u) \ln \mathcal{P}_J^{\theta}(u) du - \int \mathcal{P}_J^{\theta}(u) \ln \mathcal{P}_J^{\theta} + \Delta\theta(u) du$$

Now, observe:

$$\ln \mathcal{P}_J^{\theta} + \Delta\theta(u) = \ln \mathcal{P}_J^{\theta}(u) + \left(\frac{\nabla_{\theta} \mathcal{P}_J^{\theta}(u)}{\mathcal{P}_J^{\theta}(u)} \right)^T \Delta\theta +$$

$$\frac{1}{2} \Delta\theta^T \left(\nabla_{\theta}^2 \ln \mathcal{P}_J^{\theta}(u) \right) \Delta\theta$$

Policy gradient



(cont.)

Plugging this into the Taylor series for the KL-divergence:

$$\int \mathcal{P}_J^{\theta}(u) \ln \frac{\mathcal{P}_J^{\theta}(u)}{\mathcal{P}_J^{\theta} + \Delta\theta(u)} du \approx - \left(\int \nabla_{\theta} \mathcal{P}_J^{\theta}(u) du \right)^T \Delta\theta - \frac{1}{2} \Delta\theta^T \left(\int \mathcal{P}_J^{\theta}(u) \nabla_{\theta}^2 \ln \mathcal{P}_J^{\theta}(u) du \right) \Delta\theta$$

$$\text{But, } \int \nabla_{\theta} \mathcal{P}_J^{\theta}(u) du = \nabla_{\theta} \underbrace{\int \mathcal{P}_J^{\theta}(u) du}_{=1} = 0$$

So, only the 2nd-order term remains

Policy gradient



(cont.)

So, gathering everything together yields:

$$\lambda(\Delta\vartheta, \lambda) = J(x|\vartheta_0) + \nabla_{\vartheta} J(x|\vartheta_0) \Delta\vartheta - \frac{1}{2}\lambda \Delta\vartheta^T \left(\int_{\mathcal{U}} P_{\mathcal{U}}^{\vartheta_0}(u) \nabla_{\vartheta}^2 \ln P_{\mathcal{U}}^{\vartheta_0}(u) du \right) \Delta\vartheta + \lambda \epsilon + \mathcal{O}(\|\Delta\vartheta\|^2)$$

The term in the big parentheses is the **Fisher information!**

$$\int_{\mathcal{U}} P_{\mathcal{U}}^{\vartheta_0}(u) \nabla_{\vartheta}^2 \ln P_{\mathcal{U}}^{\vartheta_0}(u) du = I(\vartheta_0)$$

Policy gradient



(cont.)

Dropping the \mathcal{O} -term, the Lagrangian is approximately:

$$\hat{\mathcal{L}}(\Delta\vartheta, \lambda) = J(x|\vartheta_0) + \nabla_{\vartheta} J(x|\vartheta_0) \Delta\vartheta - \frac{1}{2} \lambda \Delta\vartheta^T \mathcal{I}(\vartheta_0) \Delta\vartheta + \lambda \varepsilon$$

The optimality conditions read:

$$\nabla_{\Delta\vartheta} \hat{\mathcal{L}} = \nabla_{\vartheta} J(x|\vartheta_0) - \lambda \mathcal{I}(\vartheta_0) \Delta\vartheta \stackrel{!}{=} 0$$

$$\nabla_{\lambda} \hat{\mathcal{L}} = -\Delta\vartheta^T \mathcal{I}(\vartheta_0) \Delta\vartheta + \varepsilon \stackrel{!}{=} 0$$

From the first one, we get:

$$\Delta\vartheta^* = \frac{1}{2} \mathcal{I}^{-1}(\vartheta_0) \nabla_{\vartheta} J(x|\vartheta_0)$$

Policy gradient



(cont.)

Noticing from the second equation that

$$\varepsilon = \Delta\theta^T \mathcal{L}(\theta_0) \Delta\theta$$

and plugging the $\Delta\theta^*$ into $\nabla_{\theta}\mathcal{L}$ (while noticing that \mathcal{L} is symmetric, semi-definite) yields:

$$\frac{1}{\lambda^2} \nabla_{\theta} J(x|\theta_0)^T \mathcal{L}^{-1}(\theta_0) \mathcal{L}(\theta_0) \mathcal{L}^{-1}(\theta_0) \nabla_{\theta} J(x|\theta_0) = \varepsilon$$

Policy gradient



(cont.)

Working it out, we find the optimal λ via:

$$\frac{1}{\lambda^2} \nabla_{\vartheta} J(x|\vartheta_0)^\top I^{-1}(\vartheta_0) L(\vartheta_0) L^{-1}(\vartheta_0) \nabla_{\vartheta} J(x|\vartheta_0) = \varepsilon$$

$$\frac{1}{\lambda^2} \nabla_{\vartheta} J(x|\vartheta_0)^\top L^{-1}(\vartheta_0) \nabla_{\vartheta} J(x|\vartheta_0) = \varepsilon$$

$$\Rightarrow \frac{1}{\lambda^2} = \sqrt{\frac{\varepsilon}{\| \nabla_{\vartheta} J(x|\vartheta_0) \|_{I^{-1}(\vartheta_0)}^2}} = \frac{\sqrt{\varepsilon}}{\| \nabla_{\vartheta} J(x|\vartheta_0) \|_{I^{-1}(\vartheta_0)}}$$

where $\| \cdot \|_{\bullet}$ is the weighted norm.

So,

$$\Delta \vartheta^* = \frac{\sqrt{\varepsilon} I^{-1}(\vartheta_0)}{\| \nabla_{\vartheta} J(x|\vartheta_0) \|_{I^{-1}(\vartheta_0)}} \nabla_{\vartheta} J(x|\vartheta_0)$$



(cont.)

Summarizing, the **natural gradient algorithm** reads:

- Initialize parameters ϑ_0, x_0, u_0, N , sample size
- For $i \in \mathbb{Z}_{\geq 0}$ do :

- sample state-action trajectories $\{z_j\}_j$
under $U_j \sim P_U^{\vartheta_i}(u_j | x_j)$ starting in (x_0, u_0)

- compute the sample estimates

$$\widehat{\nabla_{\vartheta} J_N(x_0 | \vartheta_i)} = \left\langle \bar{R}_N(\bar{z}_0^N) \sum_{k=0}^N \nabla_{\vartheta} \ln P_U^{\vartheta_i}(U_k | X_k), \langle \mathcal{I}(\vartheta_i) \rangle \right\rangle$$

- perform gradient ascent step:

$$\vartheta^{i+1} = \vartheta^i + \frac{\sqrt{\varepsilon} \langle \mathcal{I}^{-1}(\vartheta_i) \rangle}{\| \widehat{\nabla_{\vartheta} J(x_0 | \vartheta_i)} \|} \widehat{\nabla_{\vartheta} J(x_0 | \vartheta_i)}$$

$$\frac{\langle \mathcal{I}^{-1}(\vartheta_i) \rangle}{\| \widehat{\nabla_{\vartheta} J(x_0 | \vartheta_i)} \|}$$



Let us recall the gradient of REINFORCE:

$$\mathbb{E} \left[\bar{R}_N(\bar{Z}_0) \sum_{k=0}^N \nabla_{\theta} \ln P_{\theta}(a_k | X_k) \mid X_0 = x_0 \right]$$

We see the term $\bar{R}_N(\bar{Z}_0)$, the accumulated reward all the way along a trajectory, but it introduces some redundancy as the past rewards do not matter to the agent.

Let's try to get rid of this redundancy

Policy gradient



(cont.)

To this end, observe (omitting the $X_0 = x_0$ condition for brevity):

$$\mathbb{E} \left[\sum_{k=0}^N \nabla_\theta \ln P_\pi^{\sigma_i}(U_k | X_k) \cdot \sum_{j=0}^N \gamma^j p(X_j, U_j) \right] =$$

$$\sum_{k=0}^N \sum_{j=0}^N \mathbb{E} \left[\nabla_\theta \ln P_\pi^{\sigma_i}(U_k | X_k) \gamma^j p(X_j, U_j) \right]$$

Let's work out the summand. By tower rule,

$$\mathbb{E} \left[\nabla_\theta \ln P_\pi^{\sigma_i}(U_k | X_k) \gamma^j p(X_j, U_j) \right] =$$

$$\mathbb{E} \left[\mathbb{E} \left[\nabla_\theta \ln P_\pi^{\sigma_i}(U_k | X_k) \gamma^j p(X_j, U_j) \mid X_j, U_j \right] \right]$$

Policy gradient



$$(cont.) \quad \mathbb{E} \left[\mathbb{E} \left[\nabla_{\theta} \ln P_{\sigma}^{\theta_i}(U_k | X_k) \gamma^i p(X_j, U_j) \mid X_j, U_j \right] \right] = \underset{j < k}{\text{(assuming)}}$$

$$\mathbb{E} \left[\gamma^i p(X_j, U_j) \mathbb{E} \left[\nabla_{\theta} \ln P_{\sigma}^{\theta_i}(U_k | X_k) \mid X_j, U_j \right] \right]$$

Observe that $\mathbb{E} \left[\nabla_{\theta} \ln P_{\sigma}^{\theta_i}(U_k | X_k) \mid X_j, U_j \right] = 0$

using the same argument as in the Taylor expansion of KL - divergence (verify !)

- * The reward is then non-random under the condition of the inner expectation

Policy gradient



(cont.)

So, we conclude that

$$\mathbb{E} \left[\sum_{k=0}^N \nabla_\theta \ln P_{\pi}^{\sigma_i}(U_k | X_k) \cdot \sum_{j=0}^N \gamma^j p(X_j, U_j) \mid X_0 = x_0 \right] =$$

$$\mathbb{E} \left[\sum_{k=0}^N \nabla_\theta \ln P_{\pi}^{\sigma_i}(U_k | X_k) \cdot \sum_{j=k}^N \gamma^j p(X_j, U_j) \mid X_0 = x_0 \right] =$$

$$\mathbb{E} \left[\sum_{k=0}^N \nabla_\theta \ln P_{\pi}^{\sigma_i}(U_k | X_k) \bar{R}_N(\bar{Z}_k) \mid X_0 = x_0 \right]$$

Policy gradient



(cont.)

This greatly simplifies REINFORCE:

$$g^{i+1} = \vartheta^i + \alpha \mathbb{E} \left[\sum_{k=0}^N \nabla_{\vartheta} \ln P_{\vartheta^i}(U_k | X_k) \bar{R}_N(\bar{Z}_k) \mid X_0 = x_0 \right]$$

This principle is sometimes called „don't let
the past distract you”



Now, in the above the accumulated rewards $R_N(\bar{Z}_K^N)$.

This is one particular option. Another one is to use the so called **advantage**, which is the difference between the Q- and value functions:

$$A(x, u) := Q(x, u) - V(x)$$

Let's introduce the notation:

$\bar{R}_\infty^{V^*}(x_0, u_0)$ — the accumulated reward while taking u_0 as the first action and $V_k \sim P_{V^*}^{V^*}(u_k | x_k)$ afterwards to ∞ -horizon
(**reward-to-go**)

Policy gradient



(cont.)

Further notation:

$$Q^\vartheta(x_0, u_0) := \mathbb{E} [\bar{R}_\infty^\vartheta(x_0, u_0)]$$

$$A^\vartheta(x_0, u_0) := Q^\vartheta(x_0, u_0) - J(x_0 | \vartheta)$$

Taking ∞ -horizon in REINFORCE yields* for the gradient:

$$\mathbb{E} \left[\sum_{k=0}^{\infty} \nabla_\vartheta \ln P_\vartheta^\vartheta(U_k | X_k) \bar{R}_\infty(\bar{Z}_k) \mid X_0 = x_0 \right] =$$

$$\sum_{k=0}^{\infty} \mathbb{E} \left[\nabla_\vartheta \ln P_\vartheta^\vartheta(U_k | X_k) \bar{R}_\infty(\bar{Z}_k) \mid X_0 = x_0 \right]$$

* But be aware of ∞ -iterated integrals

Policy gradient



(cont.)

$$\sum_{k=0}^{\infty} \mathbb{E} \left[\nabla_{\theta} \ln P_{\theta}^{\sigma}(U_k | X_k) \bar{R}_{\infty}(\bar{Z}_k) \mid X_0 = x_0 \right] \quad \begin{matrix} \text{(by tower} \\ \text{rule)} \end{matrix}$$

$$= \sum_{k=0}^{\infty} \mathbb{E}_{\{U_j\}_{j=0}^k} \sim \left\{ P_{\theta}^{\sigma}(U_j | X_j) \right\}_{j=0}^k \left[\right]$$

$$\mathbb{E} \left[\nabla_{\theta} \ln P_{\theta}^{\sigma}(U_k | X_k) \bar{R}_{\infty}(\bar{Z}_k) \mid \{Z_j\}_{j=0}^k \right] \mid X_0 = x_0$$

$$= \sum_{k=0}^{\infty} \mathbb{E}_{\{U_j\}_{j=0}^k} \sim \left\{ P_{\theta}^{\sigma}(U_j | X_j) \right\}_{j=0}^k \left[\right]$$

$$\nabla_{\theta} \ln P_{\theta}^{\sigma}(U_k | X_k) \mathbb{E} \left[\bar{R}_{\infty}(\bar{Z}_k) \mid \{Z_j\}_{j=0}^k \right] \mid X_0 = x_0$$

(because the gradient is non-random under the condition of the inner expectation)

Policy gradient



(cont.)

We have:

$$\sum_{k=0}^{\infty} \mathbb{E}_{\{U_j\}_{j=0}^k} \sim \left\{ P_U^{\pi}(U_j | X_j) \right\}_{j=0}^k \quad [$$

$$\nabla_{\theta} \ln P_U^{\pi}(U_k | X_k) \mathbb{E} \left[\bar{R}_\infty(\bar{Z}_k^\infty) \middle| \{Z_j\}_{j=0}^k \right] \Big| X_0 = x_0]$$

$$= \sum_{k=0}^{\infty} \mathbb{E}_{\{U_j\}_{j=0}^k} \sim \left\{ P_U^{\pi}(U_j | X_j) \right\}_{j=0}^k \quad [$$

$$\nabla_{\theta} \ln P_U^{\pi}(U_k | X_k) \mathbb{E} \left[\bar{R}_\infty(\bar{Z}_k^\infty) \middle| Z_k \right] \Big| X_0 = x_0]$$

(by Markov property)

Policy gradient



(cont.) We have at this point:

$$\sum_{k=0}^{\infty} \mathbb{E}_{\{U_j\}_{j=0}^k} \sim \left\{ P_U^{U_i}(U_j | X_j) \right\}_{j=0}^k \left[Q^\varphi(X_k, U_k) \right]$$
$$\nabla_{\varphi} \ln P_U^\varphi(U_k | X_k) \mathbb{E} \left[\bar{R}_\infty(\bar{Z}_k) \mid Z_k \right] \mid X_0 = x_0 =$$

$$\mathbb{E} \left[\sum_{k=0}^{\infty} \nabla_{\varphi} \ln P_U^\varphi(U_k | X_k) Q^\varphi(X_k, U_k) \mid X_0 = x_0 \right]$$

which gives the ***Q*-function form** of REINFORCE.
In implementation, the ∞ -horizon is substituted
for a finite one and Q has to be
estimated



Finally, if we pick a θ -independent objective in the advantage, i.e.,

$$A^{\theta, \theta'}(x_0, u_0) := Q^\theta(x_0, u_0) - J(x_0 | \theta')$$

we get the **advantage form** of the REINFORCE gradient (with a baseline $J(x_0 | \theta')$ playing as control variate):

$$\mathbb{E} \left[\sum_{k=0}^{\infty} \nabla_{\theta} \ln P_\theta^\theta(U_k | X_k) A^{\theta, \theta'}(X_k, U_k) \mid X_0 = x_0 \right]$$

In other words, the expected value is the same, but using A may help reduce the variance. Also, the above baseline was effectively a policy, but a parametrized approximation of the value function also goes



In general, one may apply any optimization routine to policy, while gradient methods are just a particular option.

So, for instance, applying trust region methods gives the trust region policy optimization (TRPO). To define the trust region, we utilize the idea of importance sampling. Let θ_0 be the previous parameters

Policy gradient



(cont.)

Using importance sampling, we can rewrite the policy gradient as:

$$\mathbb{E} \left[\sum_{k=0}^{\infty} \nabla_{\theta} \ln P_{\pi}^{\theta}(\mathcal{U}_k | X_k) A^{\theta, \theta'}(X_k, \mathcal{U}_k) \mid X_0 = x_0 \right] =$$

$$\sum_{k=0}^{\infty} \mathbb{E} \left[\nabla_{\theta} \ln P_{\pi}^{\theta}(\mathcal{U}_k | X_k) A^{\theta, \theta'}(X_k, \mathcal{U}_k) \mid X_0 = x_0 \right] =$$

$$\sum_{k=0}^{\infty} \mathbb{E}_{\{\mathcal{U}_0\} \sim \{P_{\pi_0}^{\theta_0}\}} \left[\prod_{j=0}^k \frac{P_{\pi}^{\theta}(\mathcal{U}_j | X_j)}{P_{\pi_0}^{\theta_0}(\mathcal{U}_j | X_j)} \nabla_{\theta} \ln P_{\pi}^{\theta}(\mathcal{U}_k | X_k) A^{\theta, \theta'}(X_k, \mathcal{U}_k) \mid X_0 = x_0 \right] =$$

$$\mathbb{E}_{\{\mathcal{U}_0\} \sim \{P_{\pi_0}^{\theta_0}\}} \left[\sum_{k=0}^{\infty} \nabla_{\theta} \ln P_{\pi}^{\theta}(\mathcal{U}_k | X_k) \prod_{j=0}^k \frac{P_{\pi}^{\theta}(\mathcal{U}_j | X_j)}{P_{\pi_0}^{\theta_0}(\mathcal{U}_j | X_j)} A^{\theta, \theta'}(X_k, \mathcal{U}_k) \mid X_0 = x_0 \right],$$

from which we see that the samples are taken from another distribution, P_{π}^{θ}

Policy gradient



(cont.)

So, we got an alternative to estimate the policy gradient. To keep the variance of it under control, we restrain the gradient step size via a trust region constraint. This can again be formulated in terms of the KL-divergence, this time, an averaged one (according to Schulman et al.):

$$\mathbb{E}_{X_\infty \sim \bar{P}_x^{\theta_0}} \left[d_{KL} \left(P_v^{\theta_0}(\cdot | X_\infty) \parallel P_v^{\theta}(\cdot | X_\infty) \right) \right] \leq \epsilon ,$$

$$\text{where } \bar{P}_x^{\theta_0}(x) = \sum_{k=0}^{\infty} \gamma^k \mathbb{P}[X_k = x \mid \{U_k\}_k]$$

is the **visitation frequency** under actions sampled from $P_v^{\theta_0}$, assuming the initial state is distributed

Policy gradient



(cont.)

„Reverting“ the policy gradient back into an objective (called **surrogate advantage** in TRPO), we may write the overall optimization scheme as:

$$\max_{\vartheta} \mathbb{E}_{\{U_j\} \sim \{P_{U_j}^{\vartheta}\}} \left[\sum_{k=0}^{\infty} \prod_{j=0}^k \frac{P_U^{\vartheta}(U_j | X_j)}{P_U^{\vartheta}(U_j | X_j)} A^{\vartheta, \vartheta'}(X_k, U_k) \mid X_0 = x_0 \right]$$

$$\text{s. t. } \mathbb{E}_{X_\infty \sim \bar{P}_X^{\vartheta_0}} \left[d_{KL} \left(P_U^{\vartheta}(\cdot | X_\infty) \parallel P_U^{\vartheta}(\cdot | X_\infty) \right) \right] \leq \varepsilon$$



(cont.)

An implementation of TRPO is similar to the one of natural policy gradient plus a line search on the surrogate objective and the KL - constraint. TRPO also suggests to use conjugate gradient to estimate the product of the inverse of Fisher information with the policy gradient.



Extras:

- Proximal policy optimization (PPO) simplifies TRPO clips $\prod_{j=0}^K \frac{p_v^\pi(u_j | X_j)}{p_v^\pi(u'_j | X_j)}$ to stay within a small interval around one
- Deterministic policy gradient (DPG) uses a Markov (parametrized) Policy $\pi = K^\theta(X)$
- There is a variety of actor-critic policy gradient methods, but actor-critics are covered in another lecture
- Read on advantage estimates here:

Schulman, J., Moritz, P., Levine, S., Jordan, M., & Abbeel, P. (2015). High-dimensional continuous control using generalized advantage estimation