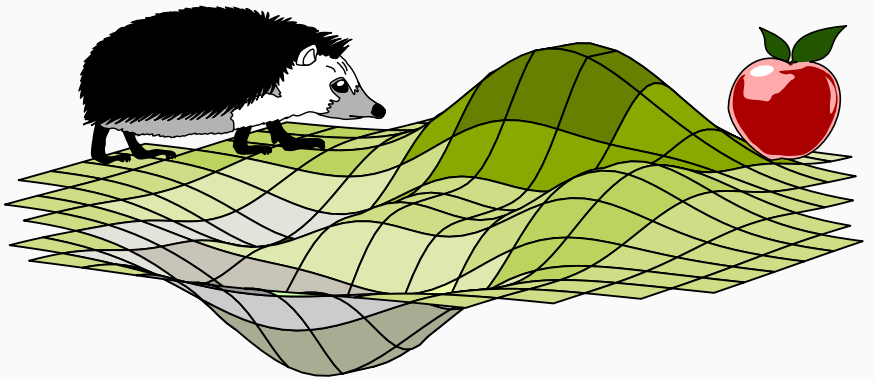
REINFORCEMENT LEARNING

POLICY GRADIENT

Pavel Osinenko



Policy gradient

Consider an environment

$$X_{k+1} \sim P_X(x_{k+2} \mid x_k, u_k)$$

$$U_k \sim P_U^{\vartheta}(u_k \mid x_k)$$

along with an ∞-horizon optimal control problem

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

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

Policy gradient

(cont.)

Let's introduce the following notation:

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

(concatenation of state-action pairs)

Thus, the environment now reads:

$$Z_{k+1} \sim P_Z^{\vartheta}(z_{k+2} \mid z_k)$$

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:

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

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

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

Policy gradient

(cont.)

Let's unwrap the objective:

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

$$= \int_{\mathcal{X}^N \times \mathcal{U}^N} \bar{R}_N(\bar{z}_0^N)\, \mathbb{P}\left[d\bar{z}_0^N \,\middle|\, X_0 = x_0\right],$$

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

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

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

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:

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

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

$$\nabla_\vartheta \left(\ln p_X(x_0) \cdot \prod_{k=0}^{N} \left(p_X(x_{k+1}|x_k,u_k) \cdot p_U^\vartheta(u_k|x_k) \right) \right) d\bar{z}_0^N$$

Policy gradient

(cont.)

Observe the following, $\forall h$:

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

So,

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

(also using the log-of-product)

$$= \mathbb{E}\left[\bar{R}_N(\bar{Z}_0^N) \sum_{k=0}^{N} \nabla_{\vartheta} \ln P_U^{\vartheta}(U_k | X_k) \,\middle|\, X_0 = x_0 \right]$$

Verify!

Policy gradient

(cont.)

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

$$\vartheta^{i+1} = \vartheta^i + \alpha \mathbb{E}\left[\bar{R}_N(\bar{Z}_0^N) \sum_{k=0}^{N} \nabla_\vartheta \ln P_U^{\vartheta_i}(U_k \mid X_k) \,\middle|\, X_0 = x_0\right]$$

$$= \vartheta^i + \alpha \mathbb{E}\left[\sum_{k=0}^{N} \nabla_\vartheta \ln P_U^{\vartheta_i}(U_k \mid X_k) \cdot \sum_{j=0}^{N} \gamma^j \rho(X_j, U_j) \,\middle|\, X_0 = x_0\right]$$

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

Policy gradient

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 „$\Delta\vartheta_k^i$" (in quotes because we are talking about distribution parameters).

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

$$d_{KL}\left(P_U^{\vartheta} \| P_U^{\vartheta+\Delta\vartheta}\right) \triangleq \int_{\mathcal{U}} P_U^{\vartheta}(u) \ln \frac{P_U^{\vartheta}(u)}{P_U^{\vartheta+\Delta\vartheta}(u)} \, du$$

Policy gradient

(cont.)

So, the algorithm would read, for instance:

$$\Delta\theta^* := \arg\max_{\Delta\theta} J(x \mid \theta_0 + \Delta\theta)$$
$$\text{s.t. } d_{KL}\left(P_J^{\theta_0} \,\|\, P_J^{\theta_0+\Delta\theta}\right) \leq \varepsilon ,$$

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:

$$\Delta\theta^* := \arg\max_{\Delta\theta} \left\{ J(x|\theta_0+\Delta\theta) - \lambda\left(d_{KL}\left(P_U^{\theta_0} \| P_U^{\theta_0+\Delta\theta}\right) - \varepsilon\right)\right\}$$

Let's denote the Lagrangian as:

$$\mathcal{L}(\Delta\theta, \lambda) = J(x|\theta_0+\Delta\theta) - \lambda\left(d_{KL}\left(P_U^{\theta_0} \| P_U^{\theta_0+\Delta\theta}\right) - \varepsilon\right),$$

in other words, the penalized objective.

The necessary optimality conditions read:

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

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

Policy gradient

(cont.)

So, we need to find the respective gradients.

Let's use the Taylor expansion:

- for the objective:

$$J(x|\vartheta_0+\Delta\vartheta) = J(x|\vartheta_0) + \nabla_\vartheta J(x|\vartheta_0)\Delta\vartheta + \mathcal{O}(\|\Delta\vartheta\|^2)$$

- for the KL-divergence:

$$d_{KL}\left(P_U^{\vartheta_0} \| P_U^{\vartheta_0+\Delta\vartheta}\right) = 0 + \langle 1\text{st order}\rangle + \langle 2\text{nd order}\rangle + \mathcal{O}(\|\Delta\vartheta\|^3)$$

Policy gradient

(cont.)

Let's first work out the KL-divergence.
Recall the formula and expand it as:

$$\int_{\mathcal{U}} P_{\mathcal{U}}^{\theta_0}(u) \ln \frac{P_{\mathcal{U}}^{\theta_0}(u)}{P_{\mathcal{U}}^{\theta_0+\Delta\theta}(u)} du = \int_{\mathcal{U}} P_{\mathcal{U}}^{\theta_0}(u) \ln P_{\mathcal{U}}^{\theta_0}(u) du - \int_{\mathcal{U}} P_{\mathcal{U}}^{\theta_0}(u) \ln P_{\mathcal{U}}^{\theta_0+\Delta\theta}(u) du$$

Now, observe:

$$\ln P_{\mathcal{U}}^{\theta_0+\Delta\theta}(u) = \ln P_{\mathcal{U}}^{\theta_0}(u) + \left(\frac{\nabla_\theta P_{\mathcal{U}}^{\theta_0}(u)}{P_{\mathcal{U}}^{\theta_0}(u)}\right)^T \Delta\theta + \frac{1}{2}\Delta\theta^T \left(\nabla_\theta^2 \ln P_{\mathcal{U}}^{\theta_0}(u)\right) \Delta\theta$$

Policy gradient

(cont.)

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

$$\int_U P_U^{\theta_0}(u) \ln \frac{P_U^{\theta_0}(u)}{P_U^{\theta_0+\Delta\theta}(u)} du \approx -\left(\int_U \nabla_\theta P_U^{\theta_0}(u) du\right)^T \Delta\theta -$$

$$\frac{1}{2}\lambda \Delta\theta^T \left(\int_U P_U^{\theta_0}(u) \nabla_\theta^2 \ln P_U^{\theta_0}(u) du\right) \Delta\theta$$

But, $$\int_U \nabla_\theta P_U^{\theta_0}(u) du = \nabla_\theta \underbrace{\int_U P_U^{\theta_0}(u) du}_{=1} = 0$$

So, only the 2nd-order term remains

Policy gradient

(cont.)

So, gathering everything together yields:

$$\mathcal{L}(\Delta\vartheta, \lambda) = J(x|\vartheta_0) + \nabla_\vartheta J(x|\vartheta_0)\Delta\vartheta - \frac{1}{2}\lambda\Delta\vartheta^T\left(\int_U P_U^{\vartheta_0}(u)\nabla_\vartheta^2 \ln P_U^{\vartheta_0}(u)du\right)\Delta\vartheta + \lambda\varepsilon + \mathcal{O}(\|\Delta\vartheta\|^2)$$

The term in the big parentheses is the Fisher information!

$$\int_U P_U^{\vartheta_0}(u)\nabla_\vartheta^2 \ln P_U^{\vartheta_0}(u)du = \mathcal{I}(\vartheta_0)$$

Policy gradient

(cont.)

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

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

The optimality conditions read:

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

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

From the first one, we get:

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

Policy gradient

(cont.)

Noticing from the second equation that

$$\varepsilon = \Delta\vartheta^T \mathcal{I}(\vartheta_0) \Delta\vartheta$$

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

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

Policy gradient

(cont.)

Working it out, we find the optimal λ via:

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

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

$$\Rightarrow \frac{1}{\lambda^*} = \sqrt{\frac{\varepsilon}{\| \nabla_\theta J(x|\theta_0) \|^2_{\mathcal{I}^{-1}(\theta_0)}}} = \frac{\sqrt{\varepsilon}}{\| \nabla_\theta J(x|\theta_0) \|_{\mathcal{I}^{-1}(\theta_0)}}$$

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

So,

$$\Delta\theta^* = \frac{\sqrt{\varepsilon}\, \mathcal{I}^{-1}(\theta_0)}{\| \nabla_\theta J(x|\theta_0) \|_{\mathcal{I}^{-1}(\theta_0)}} \nabla_\theta J(x|\theta_0)$$

Policy gradient

(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

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

  - perform gradient ascent step:

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

Policy gradient

Let us recall the gradient of REINFORCE:

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

We see the term $\bar{R}_N(\bar{Z}_0^N)$, 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):

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

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

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

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

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

Policy gradient

(cont.)

$$\mathbb{E}\left[\mathbb{E}\left[\nabla_\theta \ln P_\theta^{\sigma_i}(U_k|X_k)\,\gamma^i \rho(X_j,U_j)\,\middle|\,X_j,U_j\right]\right] = \left(\begin{matrix}\text{assuming}\\ j<k\end{matrix}\right)^*$$

$$\mathbb{E}\left[\gamma^i \rho(X_j,U_j)\,\mathbb{E}\left[\nabla_\theta \ln P_\theta^{\sigma_i}(U_k|X_k)\,\middle|\,X_j,U_j\right]\right]$$

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

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

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

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

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

Policy gradient

(cont.)

This greatly simplifies REINFORCE:

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

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

Policy gradient

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 $U_k \sim P_U^v(u_k|x_k)$ afterwards to ∞-horizon (**reward-to-go**)

Policy gradient

(cont.)

Further notation:

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

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

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

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

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

* But be aware of ∞-iterated integrals

Policy gradient

(cont.)

$$\sum_{k=0}^{\infty} \mathbb{E}\left[\nabla_\theta \ln P_U^\theta(U_k \mid X_k)\, \bar{R}_\infty(\bar{Z}_k^\infty) \,\middle|\, X_0 = x_0\right] \qquad \text{(by tower rule)}$$

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

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

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

Policy gradient

(cont.)

We have:

$$\sum_{k=0}^{\infty} \mathbb{E}_{\{U_j\}_{j=0}^{k} \sim \{P_U^{\theta_i}(U_j \mid X_j)\}_{j=0}^{k}} \Big[\nabla_\theta \ln P_U^{\theta}(U_k \mid X_k) \, \mathbb{E}\Big[\bar{R}_\infty(\bar{Z}_k^\infty) \,\Big|\, \{Z_j\}_{j=0}^{k} \Big] \,\Big|\, X_0 = x_0 \Big]$$

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

(by Markov property)

Policy gradient

(cont.) We have at this point:

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

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

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

Policy gradient

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

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

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

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

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

Policy gradient

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_o be the previous parameters

Policy gradient

(cont.)

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

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

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

$$\sum_{k=0}^{\infty} \mathbb{E}_{\{U_\bullet\}\sim\{P_{U_\bullet}^{\theta_0}\}}\left[\prod_{j=0}^{k} \frac{P_U^\theta(U_j|X_j)}{P_U^{\theta_0}(U_j|X_j)} \nabla_\theta \ln P_U^\theta(U_k|X_k)\, A^{\theta,\theta'}(X_k,U_k) \,\middle|\, X_0 = x_0\right] =$$

$$\mathbb{E}_{\{U_\bullet\}\sim\{P_{U_\bullet}^{\theta_0}\}}\left[\sum_{k=0}^{\infty} \nabla_\theta \ln P_U^\theta(U_k|X_k) \prod_{j=0}^{k} \frac{P_U^\theta(U_j|X_j)}{P_U^{\theta_0}(U_j|X_j)} A^{\theta,\theta'}(X_k,U_k) \,\middle|\, X_0 = x_0\right],$$

from which we see that the samples are taken from another distribution, $P_U^{\theta_0}$

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.):

$$\mathbb{E}_{X_\infty \sim \bar{P}_X^{\vartheta_0}} \left[d_{KL} \left(P_U^{\vartheta_0}(\bullet \mid X_\infty) \,\middle\|\, P_U^{\vartheta}(\bullet \mid X_\infty) \right) \right] \leq \varepsilon ,$$

where $\bar{P}_X^{\vartheta_0}(x) = \sum_{k=0}^{\infty} \gamma^k \mathbb{P}\left[X_k = x \,\middle|\, \{U_k\}_k \right]$

is the visitation frequency under actions sampled from $P_U^{\vartheta_0}$, 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:

$$\max_{\vartheta} \; \mathbb{E}_{\{U_\bullet\}\sim\{P^{\vartheta_0}_{U_\bullet}\}}\left[\sum_{k=0}^{\infty}\prod_{j=0}^{k}\frac{P^{\vartheta}_{U}(U_j|X_j)}{P^{\vartheta_0}_{U}(U_j|X_j)}A^{\vartheta,\vartheta'}(X_k,U_k)\,\middle|\,X_0=x_0\right]$$

$$\text{s.t.}\;\; \mathbb{E}_{X_\infty\sim\bar{P}^{\vartheta_0}_{X}}\left[d_{KL}\left(P^{\vartheta_0}_{U}(\bullet|X_\infty)\,\middle\|\,P^{\vartheta}_{U}(\bullet|X_\infty)\right)\right]\leq\varepsilon$$

Policy gradient

(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.

Policy gradient

Extras:

- Proximal policy optimization (PPO) simplifies TRPO clips $\prod_{j=0}^{k} \frac{P_U^{\vartheta}(U_j|X_j)}{P_U^{\vartheta_0}(U_j|X_j)}$ to stay within a small interval around one
- Deterministic policy gradient (DPG) uses a Markov (parametrized) policy $U = K^{\vartheta}(X)$
- 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