CS294-40 Learning for Robotics and Control Lecture 16 - 10/20/2008

Policy Gradient

Lecturer: Pieter Abbeel *Scribe: Jan Biermeyer*

1 Recap

Recall:

$$U(\theta) = \mathbb{E}[\sum_{t=0}^{H} R(s_t, a_t); \pi_\theta] = \mathbb{E}[R(\tau); \pi_\theta] \tag{1}$$

Here τ is a sample path of states and actions, $s_0, a_0, \ldots, s_H, a_H$.
Example policy π_θ:

$$a \in \{0, 1\}, \pi_\theta(0|s_t) = \frac{e^{-\theta^\top \phi(s_t)}}{1 + e^{-\theta^\top \phi(s_t)}} \tag{2}$$

2 Policy Search

$$\max_\theta U(\theta) = \max_\theta \mathbb{E}\left[R(\tau); \pi_\theta\right] \tag{3}$$

$$= \max_\theta \sum_\tau P(\tau; \theta) R(\tau) \tag{4}$$

taking the gradient w.r.t. θ gives

$$\nabla_\theta U(\theta) = \nabla_\theta \sum_\tau P(\tau; \theta) R(\tau) \tag{5}$$

$$= \sum_\tau \nabla_\theta P(\tau; \theta) R(\tau) \tag{6}$$

$$= \sum_\tau \frac{P(\tau; \theta)}{P(\tau; \theta)} \nabla_\theta P(\tau; \theta) R(\tau) \tag{7}$$

$$= \sum_\tau P(\tau; \theta) \frac{\nabla_\theta P(\tau; \theta)}{P(\tau; \theta)} R(\tau) \tag{8}$$

$$= \sum_\tau P(\tau; \theta) \nabla_\theta \log P(\tau; \theta) R(\tau) \tag{9}$$

Approximate with the empirical estimate for m sample paths under policy π_θ:

$$\hat{g} = \frac{1}{m} \sum_{i=1}^{m} \nabla_\theta \log P(\tau^{(i)}; \theta) R(\tau^{(i)}) \tag{10}$$

$$\nabla_\theta \log P(\tau^{(i)};\theta) = \nabla_\theta \log \left[\prod_{t=0}^{H} \underbrace{P(s_{t+1}^{(i)}|s_t^{(i)}, a_t^{(i)})}_{\text{dynamics model}} \cdot \underbrace{\pi_\theta(a_t^{(i)}|s_t^{(i)})}_{\text{policy}} \right] \tag{11}$$

$$= \nabla_\theta \left[\sum_{t=0}^{H} \log P(s_{t+1}^{(i)}|s_t^{(i)}, a_t^{(i)}) + \sum_{t=0}^{H} \log \pi_\theta(a_t^{(i)}|s_t^{(i)}) \right] \tag{12}$$

$$= \nabla_\theta \sum_{t=0}^{H} \log \pi_\theta(a_t^{(i)}|s_t^{(i)}) \tag{13}$$

$$= \sum_{t=0}^{H} \underbrace{\nabla_\theta \log \pi_\theta(\tau_t^{(i)}|s_t^{(i)})}_{\text{no dynamics model required!!}} \tag{14}$$

Note that:

$$\sum_\tau P(\tau;\theta) = 1 \tag{15}$$

$$\Rightarrow \nabla_\theta \sum_\theta P(\tau;\theta) = 0 \tag{16}$$

$$\Leftrightarrow \sum_\tau \nabla_\theta P(\tau;\theta) = 0 \tag{17}$$

$$\Leftrightarrow \sum_\tau \frac{P(\tau;\theta)}{P(\tau;\theta)} \nabla_\theta P(\tau;\theta) = 0 \tag{18}$$

$$\Rightarrow \mathbb{E}\left[\sum_\tau P(\tau;\theta) \nabla_\theta \log P(\tau;\theta) \right] = 0 \tag{19}$$

$$\Rightarrow \mathbb{E}_\tau \left[\log P(\tau;\theta) \right] = 0 \tag{20}$$

Unbiased gradient estimate:

$$\hat{g}_j = \frac{\partial U(\theta)}{\partial \theta_j} \tag{21}$$

$$= \frac{1}{m} \sum_{i=1}^{m} \frac{\partial}{\partial \theta_j} \log P(\tau^{(i)};\theta) \cdot R(\tau^{(i)}) \tag{22}$$

$$= \frac{1}{m} \sum_{i=i1}^{m} \left[\frac{\partial}{\partial \theta_j} \log P(\tau^{(i)};\theta) \cdot R(\tau^{(i)}) - \frac{\partial}{\partial \pi_j} \log P(\tau^{(i)};\theta) \cdot b_j^{(i)} \right] \tag{23}$$

$$\tag{24}$$

$$\boxed{\Leftrightarrow \hat{g}_j = \frac{1}{m} \sum_{i=1}^{m} \frac{\partial}{\partial \theta_j} \log P(\tau^{(i)};\theta) \cdot \left(R(\tau^{(i)}) - b_j^{(i)} \right)} \tag{25}$$

$\hat{g}_j$ is an unbiased estimate, with free parameters $b_j^{(i)}$. As $b_j^{(i)}$ has to be a constant, in practice, there is typically no reason to let it depend on i, as all trajectories $\tau^{(i)}$ are (presumably) sampled i.i.d. So in practice, we typically use b_j.

$$\hat{g}_j = \frac{\partial}{\partial \theta_j} \log P(\tau^{(i)}; \theta) \cdot (R(\tau) - b_j)\,, \tag{26}$$

Wile our gradient estimates are unbiased, in that:

$$\mathbb{E}\hat{g}_j = \frac{\partial U(\theta)}{\partial \theta_j} \tag{27}$$

they are stochastic estimates and have a variance given by:

$$\mathbb{E}\left[(\hat{g}_j - \mathbb{E}\left[\hat{g}_j\right])^2\right] \tag{28}$$

We will now describe how to choose b_j such that we are minimizing the variance of our gradient estimates:

$$\min_{b_j} \mathbb{E}\left[(\hat{g}_j - \mathbb{E}\left[\hat{g}_j\right])^2\right] = \mathbb{E}\hat{g}_j^2 + \mathbb{E}\left[(\mathbb{E}\hat{g}_j)^2\right] - 2\mathbb{E}\left[\hat{g}_j - \mathbb{E}\left[\hat{g}_j\right]\right] \tag{29}$$

$$= \mathbb{E}\hat{g}_j^2 + (\mathbb{E}\hat{g}_j)^2 - 2\mathbb{E}\left[\hat{g}_j\right]\mathbb{E}\left[\hat{g}_j\right] \tag{30}$$

$$= \mathbb{E}\hat{g}_j^2 - \underbrace{(\mathbb{E}\hat{g}_j)^2}_{=\frac{\partial U(\theta)}{\partial \theta_j} - \text{independent of } b_j} \tag{31}$$

$$\min_{b_j} \mathbb{E}\hat{g}_j^2 = \min_{b_j} \mathbb{E}\left[\left(\frac{\partial}{\partial \theta_i} \log P(\tau;\theta) \cdot (R(\tau) - b_j)\right)^2\right] \tag{32}$$

$$= \min_{b_j} \mathbb{E}\left[\left(\frac{\partial}{\partial \theta_i} log(\tau;\theta)\right)^2 \cdot \left(R(\tau)^2 + b_j^2 - 2b_j R(\tau)\right)\right] \tag{33}$$

$$= \min_{b_j} \underbrace{\mathbb{E}\left[\left(\frac{\partial}{\partial \theta_i} \log P(\tau;\theta)\right)^2 \cdot R(\tau)^2\right]}_{\text{independent of } b_j} + \mathbb{E}\left[\left(\frac{\partial}{\partial \theta_i} \log P(\tau;\theta)\right)^2 \cdot b_j^2\right] \tag{34}$$

$$- 2\mathbb{E}\left[\left(\frac{\partial}{\partial \theta_i} \log P(\tau;\theta)^2\right) \cdot b_j R(\tau)\right] \tag{35}$$

$$= \min_{b_j} b_j^2 \cdot \mathbb{E}_\tau\left[\left(\frac{\partial}{\partial \theta_i} \log P(\tau;\theta)\right)^2\right] - 2b_j \mathbb{E}_\tau\left[\left(\frac{\partial}{\partial \theta_i} \log P(\tau;\theta)^2\right)^2 R(\tau)\right] \tag{36}$$

$$\frac{\partial}{\partial b_j} = 0 \Rightarrow 2b_j \mathbb{E}_\tau\left[...\right] - 2\mathbb{E}_\tau\left[...\right] = 0 \tag{37}$$

$$\boxed{\Rightarrow b_j = \frac{\mathbb{E}_\tau\left[\left(\frac{\partial}{\partial \theta_i} \log P(\tau;\theta)\right)^2 \cdot R(\tau)\right]}{\mathbb{E}_\tau\left[\left(\frac{\partial}{\partial \theta_i} \log P(\tau;\theta)\right)^2\right]}} \tag{38}$$

If we need to minimize the variance, we can compute b_j as above from samples for a good estimate.

3 Gradient descent

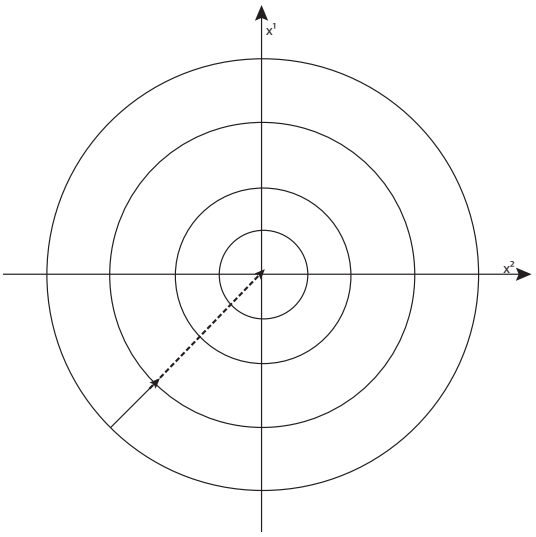

Figure 1: $f(x) = -x_1^2 - x_2^2$

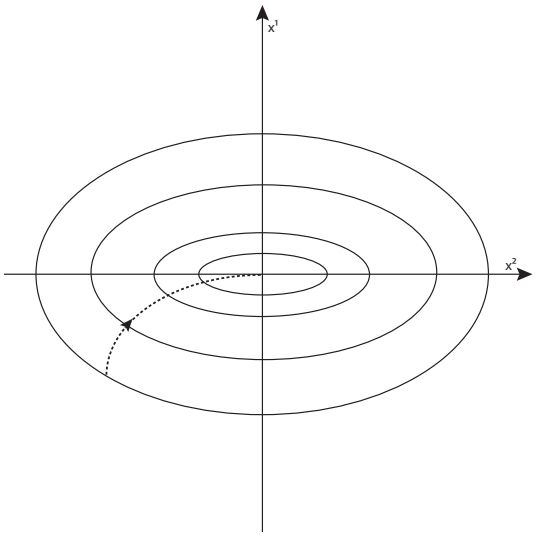

Figure 2: $\bar{f}(x) = -x_1^2 - 100x_2^2$

For function $f(x) = -x_1^2 - x_2^2$ in Figure 1, $\nabla f = -x_1 - x_2$ leads directly to the maximum. Note that $\bar{f}(x) = -x_1^2 - 100x_2^2$ in Figure 2 is essentially the same function! We just scaled the variables (e.g. could correspond to different measurements units).

Solution: look at second order methods/ approximations. Indeed, if we use Newton's method, which finds a step direction by considering the second order Taylor approximation, the resulting step directions always leads us directly to the minimum.

The gradient descent directly operates in the parameter space θ. However the same policy class π_θ can often be represented by various different parameterizations. Different parameterizations will lead to different gradients. The natural gradient method, described below, intends to get around this issue, by more directly optimizing in terms of the policy class (and distances within the policy class, i.e., distances between probability distributions) rather than the original parameters.

4 Natural gradients

We can approximate the derivative

$$\frac{\partial f}{\partial \theta_i} \approx \frac{f(\theta_i^0 + \Delta) - f(\theta_i^0 - \Delta)}{2\Delta} \tag{39}$$

where Δ is the "delta-width". Note that the gradient computation normalizes by the distance traveled in θ space. However, θ is often an arbitrary way to index into the policy class, so it might be more natural to divide by a distance that is defined directly on the policy class, rather than on θ.

For example, rather than dividing by 2Δ, we could consider dividing by

$$KL(P_{\theta_i^0 + \Delta} || P_{\theta_i^0 - \Delta}) \tag{40}$$

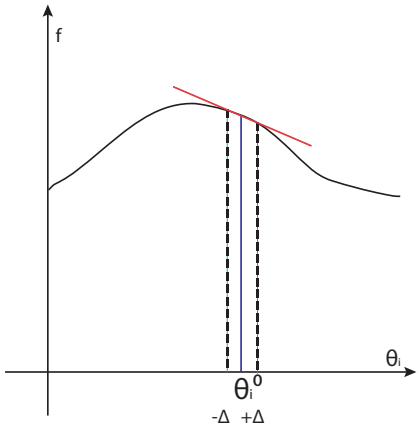

Figure 3: $f(\theta)$

The natural gradient essentially generalizes this idea, and finds the steepest ascent direction, when normalizing by a distance metric operating in the policy class space directly, rather than in the "arbitrary" θ space:

$$g_N = \arg \max_{\partial\theta_i^0 : ||\delta\theta_i^0|| = \epsilon} \frac{f(\theta_i^0 + \delta\theta_i^0) - f(\theta_i^0 - \delta\theta_i^0)}{distance(\theta_i^0 + \delta\theta_i^0, \theta_i^0 - \delta\theta_i^0)} \tag{41}$$

When ϵ is "really small," we can approximate the function f with a first-order Taylor expansion, and the distance metric by a 2nd order Taylor expansion:

$$f(\theta + \delta\theta) \approx f(\theta) + g_\theta^T \delta\theta \tag{42}$$

$$distance(\theta + \delta\theta, \theta - \delta\theta) \approx \sqrt{(\theta + \delta\theta - (\theta - \delta\theta))^\top G_\theta (\theta + \delta\theta - (\theta - \delta\theta))} = 2\sqrt{\delta\theta^T G_\theta \delta\theta} \tag{43}$$

$$\Rightarrow g_N = \arg \max_{\partial\theta : ||\delta\theta|| = \epsilon} \frac{f(\theta) + g_\theta^\top \delta\theta - (f(\delta) - g_\theta^\top \delta\theta)}{2\sqrt{\delta\theta^T G_\theta \delta\theta}} \tag{44}$$

$$= \arg \max_{\partial\theta : ||\delta\theta|| = \epsilon} \frac{g_\theta^T \delta\theta}{\sqrt{\delta\theta^T G_0 \delta\theta}} \tag{45}$$

$$\Rightarrow \boxed{\delta\theta = \alpha G_\theta^{-1} g_\theta}, \forall \alpha > 0 \tag{46}$$

The key is to pick a "clever" distance metric G_θ

$$G_\theta = I \Rightarrow g_N = \alpha \cdot g_\theta \tag{47}$$

A typical choice for distance metric between probability distributions would be a symmetric version of the KL-divergence:

$$\frac{1}{2}(KL(P\|Q) + KL(Q\|P) = \frac{1}{2}\sum_x P(x) \log \frac{P(x)}{Q(x)} + \frac{1}{2}\sum_x Q(x) \log \frac{P(x)}{Q(x)} \tag{48}$$

For this choice, we have, for θ_1 and θ_2 close enough together:

$$KL(P_{\theta_1}||P_{\theta_2}) \approx (\theta_1 - \theta_2)^T G_{\theta_1}(\theta_1 - \theta_2) \tag{49}$$

For:

$$G_\theta = E_x \left[\frac{\partial \log P_\theta(x)}{\partial \theta} \frac{\partial \log P_\theta(x)}{\partial \theta}^T \right] (\textit{Fischer information matrix}) \tag{50}$$