Lecture Note 21

1 Policy Search Methods

1.1 Off-Line "Unbiased" Gradient Descent Algorithm

Recall that we are interested in finding $\theta \in \Re^K$ such that $\nabla\lambda(\theta) = 0$, i,e, the policy parameterized policy u_θ corresponds to a local average cost minimum among the class of parameterized policies under consideration. In the previous lecture, we proposed an algorithm for performing gradient descent based on system trajectories. We assume that there is a state x^* that is recurrent under all policies u_θ. The algorithm generates a series of policies $\theta_1, \theta_2, \ldots$, which are updated whenever the system visits state x^*. The algorithm is given as follows:

1. Let θ_0 be the initial policy. Assume (for simplicity) that the initial state is $x_0 = x^*$. Let $m = 0$, $t_m = 0$.

2. Generate a trajectory $x_{t_m+1}, x_{t_m+2}, \ldots, x_{t_{m+1}}$ according to policy u_{θ_m}, where

$$t_{m+1} = \inf\{t > t_m : x_t = x^*\}.$$

3. Let

$$\begin{aligned}
\hat{h_\theta}(x_n) &= \sum_{t=n}^{t_{m+1}-1} \left[g_\theta(x_t) - \tilde{\lambda}_m\right], n = t_m + 1, \ldots, t_{m+1} - 1 \\
F_\theta(\tilde{\lambda}_m) &= \sum_{n=t_m}^{t_{m+1}-1} \left[\hat{h}_\theta(x_n) h_\theta(x_{n-1}, x_n) + \nabla g_\theta(x_n)\right] \\
\theta_{m+1} &= \theta_m - \gamma_m F_{\theta_m}(\tilde{\lambda}_m) \\
\tilde{\lambda}_{m+1} &= \tilde{\lambda}_m + \eta\gamma_m \sum_{n=t_m}^{t_{m+1}-1} \left[g_{\theta_m}(x_n) - \tilde{\lambda}_m\right]
\end{aligned}$$

4. Let $m = m + 1$, and go back to step 2.

It can be shown that this algorithm leads to a local average cost minimum, asymptotically:

Assumption 1 *Let $\rho = \{P_\theta | \theta \in \Re^k\}$ and $\bar{\rho}$ be the closure of ρ. The Markov Chain associated with P is irreducible and there exists x^* that is recurrent for every P.*

Assumption 2 *$P_\theta(x, y)$ and $g_\theta(x)$ are bounded, twice differentiable, with bounded first and second derivatives.*

Assumption 3 *There is $B < \infty$ such that $L_\theta(x, y) \leq B$ for all θ.*

Assumption 4 $\sum_{m=1}^{\infty} \gamma_m = \infty$ *and* $\sum_{m=1}^{\infty} \gamma_m^2 < \infty$.

Theorem 1 *Under Assumptions 1, 2, 3 and 4, we have*

$$\lim_{m\to\infty} \nabla\lambda(\theta_m) = 0 \quad w.p.\,1.$$

Some points about the algorithm are worth notice:

- If we had $\tilde{\lambda}_m = \lambda(\theta_m)$, $\tilde{F}_m(\tilde{\lambda}_m)$ would be an unbiased estimate of the gradient $\nabla\lambda(\theta_m)$. However, that is not the case; we only have an estimate of $\lambda(\theta_m)$. One way of getting around this difficulty would be to update the policy θ_m at a much slower rate than $\tilde{\lambda}_m$, so as to give enough time for $\tilde{\lambda}_m$ to converge to $\lambda(\theta_m)$ and generate an unbiased gradient estimate before updating the policy. However, in the algorithm described above, both θ_m and $\tilde{\lambda}_m$ are updated at the same rate, except for a constant factor η, and convergence is still guaranteed.

- The policy θ_m is updated only at visits to state x^*. This means that the algorithm can become slow when the system is large, and cycles between visits to x^* are long. In the sequel, we will look at algorithms that updated the policy at every time step.

Intuition for the proof: In order to develop some intuition for the proof of Theorem 1, we will look at the associated deterministic ODE. We have

$$\begin{aligned} \dot{\theta}_t &= -\nabla\lambda(\theta_t) - \frac{G(\theta_t)}{\mathrm{E}_\theta[T]}(\lambda(\theta_t) - \lambda_t) \\ \dot{\lambda}_t &= \eta(\lambda(\theta_t) - \lambda_t) \end{aligned}$$

We will first argue that λ_t converges, which implies that $\lambda_t - lambda(\theta_t)$ converges to zero. It follows that, asymptotically, θ_t is updated in the direction of $-\nabla\lambda(\theta_t)$, and we conclude that

$$\lim_{t\to\infty} \nabla(\lambda(\theta_t)) = 0.$$

The proof for the stochastic algorithm follows a similar argument. It turns out that neither the ODE or Lyapunov function approaches apply directly, and a customized, lengthy argument must be developed. The full proof can be found in [1].

For the convergence of λ_t, we discuss two cases:

(1) $\lambda_0 \geq \lambda(\theta_0)$

In this case, we first argue that $\lambda_t \geq \lambda(\theta_t)$. Indeed, suppose $\lambda_0 = \lambda(\theta_{t_0})$ for some t_0. Then either $\nabla\lambda(\theta_{t_0}) = 0$, and the ODE reaches an equilibrium, or $\dot{\lambda}(\theta_{t_0}) < 0$ and $\dot{\lambda}_{t_0} = 0$. We conclude that $\lambda_t \geq \lambda(\theta_t)$, $\forall t$.

From the above discussion, we conclude that λ_t is nonincreasing and bounded. Therefore, λ_t converges.

(2) $\lambda_0 < \lambda(\theta_0)$

We have two possible situations:

(i) $\lambda_t < \lambda(\theta_t)$, $\forall t$

In this case, $\Rightarrow \lambda_t$ is nondecreasing and bounded, therefore it converges.

(ii) $\lambda_{t_0} = \lambda(\theta_{t_0})$ for some t_0 In this case, we are back to case (1).

We conclude that λ_t converges, and thus $(\lambda(\theta_t) - \lambda_t) \to 0$. Therefore, $\dot{\theta}_t \to -\nabla\lambda(\theta_t)$ asymptotically, and $\nabla\lambda(\theta_t) \to 0$.

1.2 Online "Unbiased" Gradient Descent Algorithm

We now develop a version of gradient descent where the policy is updated in every time step, rather than only at visits to state x^*. The algorithm has the advantage of being simpler and potentially faster.

First note that F_m can be computed incrementally between visits to state x^*:

$$\begin{aligned}
F_m(\theta,\tilde{\lambda}) &= \sum_{n=t_m}^{t_{m+1}-1}\left[\hat{h}_\theta(x_n)L_\theta(x_{n-1},x_n)+\nabla g_\theta(x_n)\right] \\
&= \nabla g_\theta(x^*)+\sum_{n=t_m+1}^{t_{m+1}-1}\left[\hat{h}_\theta(x_n)L_\theta(x_{n-1},x_n)+\nabla g_\theta(x_n)\right] \\
&= \nabla g_\theta(x^*)+\sum_{n=t_m+1}^{t_{m+1}-1}\left[\sum_{k=n}^{t_{n+1}-1}\left(g_\theta(x_k)-\tilde{\lambda}\right)L_\theta(x_{k-1},x_k)+\nabla g_\theta(x_n)\right] \\
&= \nabla g_\theta(x^*)+\sum_{n=t_m+1}^{t_{m+1}-1}\left[\nabla g_\theta(x_n)+\sum_{t_m+1}^{n}L_\theta(x_{k-1},x_k)(g_\theta(x_n)-\tilde{\lambda})\right] \\
&= \nabla g_\theta(x^*)+\sum_{n=t_m+1}^{t_{m+1}-1}\left[\nabla g_\theta(x_n)+\left(g_\theta(x_n)-\tilde{\lambda}\right)z_n\right]
\end{aligned}$$

where

$$z_n=\sum_{k=t_m+1}^{n}L_\theta(x_{k-1},x_k)\Rightarrow z_n=z_{n-1}+L_\theta(x_{n-1},x_n)$$

. This suggests the following **Online Algorithm**:

$$\begin{aligned}
\theta_{k+1} &= \theta_k-\gamma_k\left[\nabla g_\theta(x_k)+\left((g_{\theta_k}(x_k)-\tilde{\lambda}_k)z_k\right)\right] \\
z_{k+1} &= \begin{cases} 0 & \text{if } x_{k+1}=x^* \\ z_k+L_\theta(x_k,x_{k+1}) & \text{otherwise}\end{cases}
\end{aligned}$$

Assumption 5 *Let $\mathcal{P}=\{P_\theta:\theta\in\Re^k\}$ and $\bar{\mathcal{P}}$ be the closure of $\mathcal{P}$. Then there exists N_0 such that, $\forall(P_1,P_2,\ldots,P_{N_0})$, $P_i\in\bar{\mathcal{P}}$, $\forall x$,*

$$\sum_{n=1}^{N_0}\prod_{l=1}^{n}P_l(x,x^*)>0.$$

Assumption 6 *$\sum\gamma_k=\infty$, $\sum\gamma_k^2<\infty$, $\gamma_k\le\gamma_{k-1}$, and $\sum_{k=n}^{n+t}(\gamma_n-\gamma_k)\le At^P\gamma_n^P$ for some A and P.*

Theorem 2 *Under Assumptions 1-6, we have*

$$\nabla\lambda(\theta_k)\to 0,\quad w.p.1.$$

The idea behind the proof of Theorem 2 is that, due to the assumptions on the step sizes γ_k (Assumption 6, eventually changes in the policy θ_m made between two consecutive visits to state x^* are negligible, and the algorithm behaves very similarly to the offline version. Assumption 5 is required in order to guarantee that the time between visits to state x^* remains small, even as the policy is not stationary.

1.3 Biased Gradient Estimation

In both the offline and online "unbiased" gradient descent algorithms, the variance of the estimates depends on the variance of the times between visits to state x^*, which can be very large depending on the system. We now look at a different algorithm, which is aimed at developing estimates $\tilde{\nabla}\lambda(\theta)$ with smaller variance. The decrease in variance is traded against a potential bias in the estimate, i.e., we have $\mathrm{E}\tilde{\nabla}\lambda(\theta) \neq \nabla\lambda(\theta)$. Note that a small amount of bias may still be acceptable since it should suffice to have estimates that have positive inner product with the true gradient, in order for the algorithm to converge:

$$\langle \mathrm{E}\tilde{\nabla}\lambda(\theta), \nabla\lambda(\theta)\rangle > 0.$$

We generate one such biased estimate $\tilde{\nabla}\lambda(\theta)$ based on a discounted-cost approximation. Let

$$J_{\theta,\alpha}(x) = \mathrm{E}_\theta\left[\sum_{t=0}^{\infty} \alpha^t g(x_t) | x_0 = x\right]$$

Then we have

Theorem 3

$$\nabla\lambda(\theta) = (1-\alpha)\nabla\pi_\theta^T J_{\theta,\alpha} + \underbrace{\alpha\pi_\theta^T \nabla P_\theta J_{\theta,\alpha}}_{\nabla_\alpha\lambda(\theta)}$$

Proof: We have $J_{\theta,\alpha} = g + \alpha P_\theta J_{\theta,\alpha}$. Then

$$\nabla\lambda(\theta) = \nabla\pi_\theta^T g = \nabla\pi_\theta^T [J_{\theta,\alpha} - \alpha P_\theta J_{\theta,\alpha}].$$

Since $\pi_\theta^T = \pi_\theta^T P$, we have

$$\nabla\pi_\theta^T = \nabla\pi_\theta^T P_\theta + \pi_\theta^T \nabla P_\theta.$$

Hence,

$$\begin{aligned}
\nabla\lambda(\theta) &= \nabla\pi_\theta^T J_{\theta,\alpha} - \alpha\nabla\pi_\theta^T P_\theta J_{\theta,\alpha} \\
&= \nabla\pi_\theta^T J_{\theta,\alpha} - \alpha\nabla\pi_\theta^T J_{\theta,\alpha} + \alpha\pi_\theta^T \nabla P_\theta J_{\theta,\alpha} \\
&= (1-\alpha)\nabla\pi_\theta^T J_{\theta,\alpha} + \alpha\pi_\theta^T \nabla P_\theta J_{\theta,\alpha}
\end{aligned}$$

□

The following theorem shows that $\nabla_\alpha\lambda(\theta)$ can be used as an approximation to $\nabla\lambda(\theta)$, if α is reasonably close to one.

Theorem 4 *Let* $\nabla_\alpha\lambda(\theta) = \alpha\pi_\theta^T \nabla P_\theta J_{\theta,\alpha}$. *Then*

$$\lim_{\alpha\to 1} \nabla_\alpha\lambda(\theta) = \nabla\lambda(\theta)$$

Proof: We have

$$J_{\theta,\alpha} = \frac{\lambda(\theta)}{1-\alpha} e + h_\theta + O(|1-\alpha|).$$

Therefore,

$$\begin{aligned}(1-\alpha)\nabla\pi_\theta^T J_{\theta,\alpha} &= (1-\alpha)\nabla\pi_\theta^T\left[\frac{\lambda(\theta)}{1-\alpha}e + h_\theta + O(|1-\alpha|)\right] \\ &= (1-\alpha)\nabla\pi_\theta^T\frac{\lambda(\theta)}{1-\alpha}e + \underbrace{(1-\alpha)\nabla\pi_\theta^T(h_\theta + O(|1-\alpha|))}_{\to 0 \text{ as } \alpha\to 1} \\ &= \lambda(\theta)\nabla\pi_\theta^T e + O(|1-\alpha|)\end{aligned}$$

But $\pi_\theta^T e = 1$, we have $\nabla\pi_\theta^T e = 0$. Therefore,

$$(1-\alpha)\nabla\pi_\theta^T J_{\theta,\alpha} = 0 + O(|1-\alpha|) \to 0 \text{ as } \alpha \to 1.$$

□

If we want to use $\nabla_\alpha\lambda(\theta)$ instead of $\nabla\lambda(\theta)$, a simulation-based algorithm will compute $\hat{J}_{\theta,\alpha}$ instead of $\hat{h}_\theta$.

We have

$$Var(\hat{h}_\theta) \approx O(\mathrm{E}[T^2]) \quad \text{and} \quad Var\left(\hat{J}_{\theta,\alpha}\right) = O\left(\frac{1}{(1-\alpha)^2}\right)$$

However, using $\nabla_\alpha\lambda(\theta)$, we have a bias $O(\mathrm{E}[T](1-\alpha))$.

Based on the previous discussion, we can generate an algorithm for estimating $\nabla_\alpha\lambda(\theta)$ using the same ideas from the offline unbiased gradient descent algorithm. Indeed, consider the following algorithm, where the policy is held fixed:

$$\begin{aligned}\triangle_{k+1} &= \triangle_k + \frac{1}{k+1}\left(g(x_k)z_{k+1} - \triangle_k\right) \\ z_{k+1} &= \alpha z_k + L_\theta(x_k, x_{k+1})\end{aligned}$$

Then it can be shown that $\triangle_k \to \nabla_\alpha\lambda(\theta)$, if the policy is held fixed. The gradient estimate $\triangle_k$ can be used for updating the policy in an offline or online fashion, just as with the unbiased gradient descent algorithms.

Assumption 7 *1. unique π_θ for each θ*

2. $|g(x_k)| \le B, \forall x$

3. $|L_\theta(x,y)| \le B, \forall x, y$

Theorem 5 *Under Assumption 7, we have*

$$\lim_{k\to\infty} \triangle_k \to \nabla_\alpha\lambda(\theta), \quad w.p.1.$$

References

[1] P. Marbach and J.N. Tsitsiklis. Simulation-based optimization of Markov reward processes. *IEEE Transactions on Automatic Control*, 46(2):191–209, 2001.