CS 287: Advanced Robotics
Fall 2009

Lecture 18: Policy search

Pieter Abbeel
UC Berkeley EECS
### Policy gradient

<table>
<thead>
<tr>
<th></th>
<th>Deterministic</th>
<th>Stochastic</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Known Dynamics</td>
<td>Unknown Dynamics</td>
</tr>
<tr>
<td>Analytical</td>
<td>OK</td>
<td>N/A</td>
</tr>
</tbody>
</table>
|                      | Taking derivatives--- potentially time consuming and error-prone | OK
|                      | N/A           | OK                  |
|                      | OK            | Often computationally impractical |
|                      | N/A           | Same as known dynamics, but no fixing of random seed. |
|Finite differences    | OK            | OK                  |
|                      | Sometimes computationally more expensive than analytical | OK
|                      | N/A           | N(A^{1/2}) [1]      |
|Likelihood ratio method| OK            | OK                  |
|                      | OK            | OK                  |
|                      | N(A^{1/2}) [1] | Same as known dynamics, but no fixing of random seed. |


### Likelihood ratio method

- **Assumption:**
  - Stochastic policy $\pi_\theta(u_t | s_t)$

- **Stochasticity:**
  - **Required for the methodology**
  - **+** Helpful to ensure exploration
  - **-** Optimal policy within the policy class is not always stochastic (though it can be!!)
We let $\tau$ denote a state-action sequence $s_0, u_0, \ldots, s_H, u_H$. We overload notation: $R(\tau) = \sum_{t=0}^{H} R(s_t, u_t)$.

\[ U(\theta) = E[\sum_{t=0}^{H} R(s_t, u_t); \pi_{\theta}] = \sum_{\tau} P(\tau; \theta) R(\tau) \]

Our goal is to find $\theta$:

\[ \max_{\theta} U(\theta) = \max_{\theta} \sum_{\tau} P(\tau; \theta) R(\tau) \]

**Likelihood ratio method**

\[ U(\theta) = \sum_{\tau} P(\tau; \theta) R(\tau) \]

Taking the gradient w.r.t. $\theta$ gives

\[ \nabla_{\theta} U(\theta) = \sum_{\tau} \nabla_{\theta} P(\tau; \theta) R(\tau) \]

\[ = \sum_{\tau} P(\tau; \theta) \frac{\partial}{\partial \theta} \log P(\tau; \theta) R(\tau) \]

\[ = \sum_{\tau} P(\tau; \theta) \nabla_{\theta} \log P(\tau; \theta) R(\tau) \]

Approximate with the empirical estimate for $m$ sample paths under policy $\pi_{\theta}$:

\[ \nabla_{\theta} U(\theta) \approx \frac{1}{m} \sum_{i=1}^{m} \nabla_{\theta} \log P(\tau(i); \theta) R(\tau(i)) \]

**Likelihood ratio method derivation**

\[ \nabla_{\theta} \log P(\tau(i); \theta) = \nabla_{\theta} \log \left( \prod_{t} P(s_{t+1}|s_t, u_t; \theta) \pi_{\theta}(u_t|s_t) \right) \]

\[ = \nabla_{\theta} \log \left( \prod_{t} P(s_{t+1}|s_t, u_t; \theta) \right) + \sum_{t} \log \pi_{\theta}(u_t|s_t) \]

\[ = \sum_{t} \nabla_{\theta} \log P(s_{t+1}|s_t, u_t; \theta) + \sum_{t} \log \pi_{\theta}(u_t|s_t) \]

\[ = \sum_{t} \nabla_{\theta} \log P(s_{t+1}|s_t, u_t; \theta) \]
Likelihood ratio method: result recap

The following expression provides us with an unbiased estimate of the gradient, and we can compute it without access to a dynamics model:

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

Here:

\[ \nabla_{\theta} \log P(\tau^{(i)}; \theta) = \sum_{t=0}^{H} \nabla_{\theta} \log \pi_{\theta}(u^{(i)}_{t} | s^{(i)}_{t}) \]

no dynamics model required!!

Unbiased means:

\[ E[\hat{g}] = \nabla_{\theta} U(\theta) \]

Likelihood ratio method in practice

- As formulated thus far: yes, unbiased estimator, but very noisy, hence would take very long

- Set of critical fixes that have led to real-world practicality:
  - Add a free parameter to the estimator called “baseline” and set it such that the variance of the estimator is minimized
  - Exploit temporal structure + incorporate value function estimates (= actor-critic learning)
  - Don’t step in the direction of the gradient, follow the “natural” gradient direction instead
Consider the following scenario:
There are two envelopes, each of which has an unknown amount of money in it. You get to choose one of the envelopes. Given this is all you get to know, how should you choose?

Consider the changed scenario:
Same as above, but before you get to choose, you can ask me to disclose the amount in one of the envelopes. Without any distributional assumptions on the amounts of money, is there a strategy that could improve your expected pay-off over simply picking an envelope at random?
Envelopes riddle

- MDP:
  - horizon of 1, always start in state 0
  - Transition to state 1 or 2 according to choice made
  - Observe the reward in the visited state

- Policy
  \[ \pi_\theta(1|0) = \frac{\exp(\theta)}{1 + \exp(\theta)} \]
  \[ \pi_\theta(2|0) = \frac{1}{1 + \exp(\theta)} \]

- Choose to see an envelope’s contents according to \( \pi_\theta \)

- Perform a gradient update:
  \[ \nabla_\theta \log P(\tau = 1; \theta) R(\tau = 1) = \frac{1}{1 + \exp(\theta)} R(1) \]
  \[ \nabla_\theta \log P(\tau = 2; \theta) R(\tau = 2) = -\frac{\exp(\theta)}{1 + \exp(\theta)} R(2) \]
Perform a gradient update:

This gradient update is simply making the recently observed path more likely; and how much more likely depends on the observed R for the observed path.

\[ \nabla_{\theta} \log P(\tau = 1; \theta) R(\tau = 1) = \frac{1}{1 + \exp(\theta)} R(1) \]

\[ \nabla_{\theta} \log P(\tau = 2; \theta) R(\tau = 2) = -\frac{\exp(\theta)}{1 + \exp(\theta)} R(2) \]

This gradient update is simply making the recently observed path more likely; and how much more likely depends on the observed R for the observed path.

\[ \rightarrow \text{ rather than let it depend simply on } R, \text{ if we had a "baseline" } b \text{ which is an estimate of the expected reward under the current policy, then could update scaled by } (R-b) \text{ instead,} \]

i.e. the baseline enables updating such that better than average paths become more likely, less than average paths become less likely.

Likelihood ratio method with baseline

Gradient estimate with baseline:

\[ \hat{g} = \frac{1}{m} \sum_{i=1}^{m} \nabla_{\theta} \log P(\tau^{(i)}; \theta)(R(\tau^{(i)}) - b) \]

This will (crudely speaking) increase the log-likelihood of paths with higher than baseline reward, and decrease the log-likelihood of observed paths with lower than baseline reward.

Is this still an unbiased gradient estimate?

Unbiased means: \[ E[\hat{g}] = \nabla_{\theta} U(\theta) \]
Even with baseline, we obtain an unbiased estimate of the gradient

\[ \sum_{\tau} P(\tau; \theta) = 1 \]

\[ \Rightarrow \frac{\partial}{\partial \theta_i} \sum_{\tau} P(\tau; \theta) = 0 \]

\[ \Rightarrow \sum_{\tau} \frac{\partial}{\partial \theta_j} P(\tau; \theta) = 0 \]

\[ \Rightarrow \sum_{\tau} P(\tau; \theta) \frac{\partial}{\partial \theta_j} P(\tau; \theta) = 0 \]

\[ \Rightarrow \sum_{\tau} P(\tau; \theta) \frac{\partial}{\partial \theta_j} \log P(\tau; \theta) = 0 \]

\[ \Rightarrow \mathbb{E}_\tau \left[ \frac{\partial}{\partial \theta_j} \log P(\tau; \theta) \right] = 0 \]

\[ \Rightarrow \mathbb{E}_\tau \left[ \frac{\partial}{\partial \theta_j} \log P(\tau; \theta) b_j \right] = 0 \]

\[ \frac{\partial}{\partial \theta_j} U(\theta) = \mathbb{E}_\tau \left[ \frac{\partial}{\partial \theta_j} \log P(\tau; \theta) R(\tau) \right] \]

\[ = \mathbb{E}_\tau \left[ \frac{\partial}{\partial \theta_j} \log P(\tau; \theta)(R(\tau) - b_j) \right] \]

\[ \approx \frac{1}{m} \sum_{i=1}^{m} \left[ \frac{\partial}{\partial \theta_j} \log P(\tau^{(i)}; \theta) (R^{(i)} - b_j^{(i)}) \right]^2 \]

Natural choices for \( b \):

- Estimate of utility \( U(\theta) \)
- Choose \( b_j \) to minimize the variance of the gradient estimates.

Our gradient estimate:

\[ \hat{g}_j = \frac{\partial}{\partial \theta_j} \log P(\tau^{(i)}; \theta) \cdot (R(\tau) - b_j) \]

It is unbiased, i.e.:

\[ \mathbb{E}\hat{g}_j = \frac{\partial U(\theta)}{\partial \theta_j} \]

Its variance is given by:

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

which we would like to minimize over \( b_j \):

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

\[ = \mathbb{E}\hat{g}_j^2 + (\mathbb{E}\hat{g}_j)^2 - 2\mathbb{E}[\hat{g}_j] \mathbb{E}[\hat{g}_j] \]

\[ = \mathbb{E}\hat{g}_j^2 - (\mathbb{E}\hat{g}_j)^2 \approx \frac{\partial U(\theta)}{\partial \theta_j} \text{ independent of } b_j \]
\[
\min_{b_j} E \hat{g}_j^2 = \min_{b_j} E_r \left[ \left( \frac{\partial}{\partial \theta_j} \log P(\tau; \theta) \cdot (R(\tau) - b_j) \right)^2 \right]
\]
\[
= \min_{b_j} E_r \left[ \left( \frac{\partial}{\partial \theta_j} \log(\tau; \theta) \right)^2 \cdot (R(\tau)^2 + b_j^2 - 2b_j R(\tau)) \right]
\]
\[
= \min_{b_j} E_r \left[ \left( \frac{\partial}{\partial \theta_j} \log P(\tau; \theta) \right)^2 \cdot R(\tau)^2 \right] + E \left[ \left( \frac{\partial}{\partial \theta_j} \log P(\tau; \theta) \right)^2 \cdot b_j^2 \right]
\]
\[
- 2E \left[ \left( \frac{\partial}{\partial \theta_j} \log P(\tau; \theta)^2 \right) \cdot b_j R(\tau) \right]
\]
\[
= \min_{b_j} b_j^2 \cdot E_r \left[ \left( \frac{\partial}{\partial \theta_j} \log P(\tau; \theta) \right)^2 \right] - 2b_j E_r \left[ \left( \frac{\partial}{\partial \theta_j} \log P(\tau; \theta)^2 \right) R(\tau) \right]
\]
\[
\frac{\partial}{\partial b_j} = 0 \Rightarrow 2b_j E_r \left[ \frac{\partial}{\partial \theta_j} \log P(\tau; \theta)^2 \right] - 2E_r \left[ \frac{\partial}{\partial \theta_j} \log P(\tau; \theta)^2 \right] R(\tau) = 0
\]
\[
\Rightarrow b_j = \frac{E_r \left[ \left( \frac{\partial}{\partial \theta_j} \log P(\tau; \theta) \right)^2 \cdot R(\tau) \right]}{E_r \left[ \left( \frac{\partial}{\partial \theta_j} \log P(\tau; \theta) \right)^2 \right]}
\]

\[\rightarrow\] Could estimate optimal baseline from samples.

---

**Exploiting temporal structure**

Our gradient estimate:

\[
\hat{g}_j = \frac{1}{m} \sum_{i=1}^{m} \left( \frac{\partial}{\partial \theta_j} \log P(\tau^{(i)}; \theta) \right) \left( R(\tau^{(i)}) - b_j \right),
\]
\[
= \frac{1}{m} \sum_{i=1}^{m} \left( \sum_{t=0}^{H-1} \frac{\partial}{\partial \theta_j} \log \pi_\theta(u_t^{(i)} | s_t^{(i)}) \sum_{t=0}^{H-1} R(s_t^{(i)}, u_t^{(i)}) \right) - b_j)
\]

Future actions do not depend on past rewards (assuming a fixed policy). This can be formalized as

\[
E \left[ \frac{\partial}{\partial \theta_j} \log \pi_\theta(u_k | s_k) R(s_k, u_k) \right] = 0 \quad \forall k < t
\]

Removing these terms with zero expected value from our gradient estimate we obtain:

\[
\hat{g}_j = \frac{1}{m} \sum_{i=1}^{m} \sum_{t=0}^{H-1} \frac{\partial}{\partial \theta_j} \log \pi_\theta(u_t^{(i)} | s_t^{(i)}) \left( \sum_{k=t}^{H-1} R(s_k^{(i)}, u_k^{(i)}) - b_j \right)
\]
Our gradient estimate:

\[
\hat{g}_j = \frac{1}{m} \sum_{i=1}^{m} \sum_{t=0}^{H-1} \frac{\partial}{\partial \theta_j} \log \pi_\theta(u_t^{(i)}|s_t^{(i)}) \left( \sum_{k=t}^{H-1} R(s_k^{(i)}, u_k^{(i)}) - b_j \right)
\]

The term \(\sum_{k=t}^{H-1} R(s_k^{(i)}, u_k^{(i)})\) is a sample based estimate of \(Q^\pi(s_t^{(i)}, u_t^{(i)})\). If we simultaneously run a temporal difference (TD) learning method to estimate \(Q^\pi\), then we could substitute its estimate for \(Q\) instead of the sample based estimate!

Our gradient estimate becomes:

\[
\hat{g}_j = \frac{1}{m} \sum_{i=1}^{m} \sum_{t=0}^{H-1} \frac{\partial}{\partial \theta_j} \log \pi_\theta(u_t^{(i)}|s_t^{(i)}) \left( \hat{Q}^\pi(s_t^{(i)}, u_t^{(i)}) - b_j \right)
\]

**Natural gradient**

- Is the gradient the correct direction?
Natural gradient

- Is the gradient the correct direction?