CS 446/ECE 449: Machine Learning

A. G. Schwing

University of Illinois at Urbana-Champaign, 2020

L30: Policy Gradient

Goals of this lecture

- More about Reinforcement Learning Techniques
- Getting to know Policy Gradient
- Understanding its relation to other methods

Recap so far: Known MDP

- To compute V^*, Q^*, π^*: use **policy/value iteration or exhaustive**
- To evaluate fixed policy π: use policy evaluation

Unknown MDP

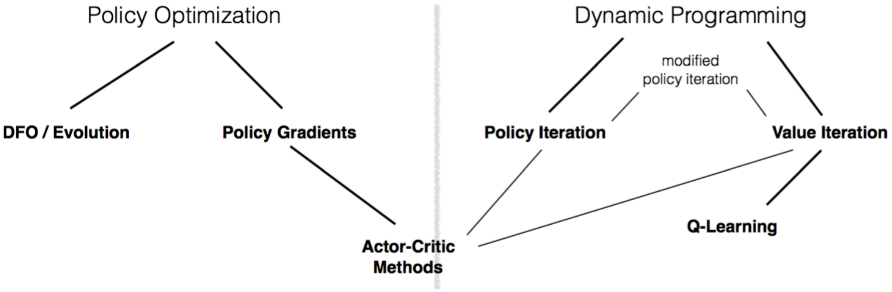
- Estimate transition probabilities using experience replay
- Q-learning

What else:

Directly optimize parametric policy $\pi_\theta(a|s)$

Why:

- π may be simpler than Q or V
- V doesn't prescribe actions: dynamics model + Bellman back-up needed
- Q requires efficient maximization: issue in continuous/high-dimensional action spaces



John Schulman & Pieter Abbeel – OpenAI + UC Berkeley

Variant: Likelihood ratio policy gradient

- Rollout, state-action sequence: $\tau = (s_0, a_0, s_1, a_1, \ldots)$
- Expected reward: $R(\tau) = \sum_t R(s_t, a_t)$

$$U(\theta) = \mathbb{E}\left[\sum_t R(s_t, a_t); \pi_\theta\right] = \sum_\tau P(\tau; \theta) R(\tau)$$

Goal:

$$\max_\theta U(\theta) = \max_\theta \sum_\tau P(\tau; \theta) R(\tau)$$

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

Related work:

- Aleksandrov, Sysoyev & Shemaneva; 1968
- Rubinstein; 1969
- Glynn; 1986
- Williams; 1992 –> Reinforce
- Baxter & Bartlett; 2001

$$\max_\theta U(\theta) = \max_\theta \sum_\tau P(\tau;\theta)R(\tau)$$

Gradient descent:

$$\begin{aligned}
\nabla_\theta U(\theta) &= \nabla_\theta \sum_\tau P(\tau;\theta)R(\tau) \\
&= \sum_\tau \nabla_\theta P(\tau;\theta)R(\tau) \\
&= \sum_\tau \frac{P(\tau;\theta)}{P(\tau;\theta)} \nabla_\theta P(\tau;\theta)R(\tau) \\
&= \sum_\tau P(\tau;\theta)\frac{\nabla_\theta P(\tau;\theta)}{P(\tau;\theta)} R(\tau) \\
&= \sum_\tau P(\tau;\theta)\nabla_\theta \log P(\tau;\theta)R(\tau)
\end{aligned}$$

Approx. with empirical estimate for sample paths under policy π_θ:

$$\nabla_\theta U(\theta) \approx \hat{g} = \frac{1}{m}\sum_{i=1}^m \nabla_\theta \log P(\tau^{(i)};\theta)R(\tau^{(i)})$$ approx. impossible w/o trick

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

Important property:

Valid even if R is discontinuous

Intuition:

- Note similarity to maximum likelihood
- Increase probability of paths τ with positive R
- Decrease probability of paths τ with negative R

No need for dynamics model:

$$
\begin{aligned}
\nabla_\theta \log P(\tau;\theta) &= \nabla_\theta \log \left[\prod_t \underbrace{P(s_{t+1}|s_t,a_t)}_{\text{dynamics model}} \underbrace{\pi_\theta(a_t|s_t)}_{\text{policy}}\right] \\
&= \nabla_\theta \left[\sum_t \log P(s_{t+1}|s_t,a_t) + \sum_t \log \pi_\theta(a_t|s_t)\right] \\
&= \nabla_\theta \sum_t \log \pi_\theta(a_t|s_t) \\
&= \sum_t \underbrace{\nabla_\theta \log \pi_\theta(a_t|s_t)}_{\text{no dynamics model required}}
\end{aligned}
$$

Consequently:

$$
\nabla_\theta \log P(\tau;\theta) = \sum_t \nabla_\theta \log \pi_\theta(a_t|s_t)
$$

From

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

to

$$\nabla_\theta U(\theta) \approx \hat{g} = \frac{1}{m}\sum_{i=1}^{m} \left(\sum_t \nabla_\theta \log \pi_\theta(a_t^{(i)}|s_t^{(i)})\right) R(\tau^{(i)})$$

Practically important:

- Baseline
- Temporal structure

Baseline: issue when $R(\tau^{(i)}) > 0$ can be fixed with

$$\nabla_\theta U(\theta) \approx \hat{g} = \frac{1}{m}\sum_{i=1}^{m} \nabla_\theta \log P(\tau^{(i)};\theta)\left(R(\tau^{(i)}) - b\right)$$

Why is subtraction of baseline b okay?

$$\begin{aligned}
\mathbb{E}\left[\nabla_\theta \log P(\tau;\theta) b\right] &= \sum_\tau P(\tau;\theta)\nabla_\theta \log P(\tau;\theta) b \\
&= \sum_\tau \nabla_\theta P(\tau;\theta) b \\
&= \nabla_\theta \left(\sum_\tau P(\tau;\theta)\right) b = 0
\end{aligned}$$

Choices of b: e.g., (others are available, e.g., Greensmith et al. (2004))

$$b = \mathbb{E}\left[R(\tau)\right] = \frac{1}{m}\sum_{i=1}^{m} R(\tau^{(i)})$$

Temporal structure:

$$\nabla_\theta U(\theta) \approx \hat{g} = \frac{1}{m}\sum_{i=1}^{m}\left(\sum_t \nabla_\theta \log \pi_\theta(a_t^{(i)}|s_t^{(i)})\right)\left(\sum_t R(s_t^{(i)}, a_t^{(i)}) - b\right)$$

Future actions don't depend on past rewards: lower variance via

$$\nabla_\theta U(\theta) \approx \hat{g} = \frac{1}{m}\sum_{i=1}^{m}\sum_t \nabla_\theta \log \pi_\theta(a_t^{(i)}|s_t^{(i)})\left(\sum_{\hat{t}\geq t} R(s_{\hat{t}}^{(i)}, a_{\hat{t}}^{(i)}) - b(s_{\hat{t}}^{(i)})\right)$$

Good choices for b:

$$b(s_t) = \mathbb{E}\left[r_t + r_{t+1} + \ldots\right]$$

Algorithm: Reinforce aka vanilla Policy Gradient

- Initial θ, b
- For iteration = 1, 2, ...
  - Collect a set of trajectories $\tau^{(i)}$ by executing policy π_θ
  - Compute reward and bias

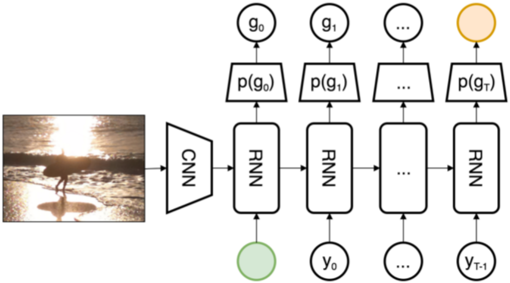
$$R_t^{(i)} = \sum_{\hat{t} \geq t} \gamma^{\hat{t}-t} r_{\hat{t}}$$

  - Re-fit the baseline b
  - Update the policy using the policy gradient estimate $\hat{g}$

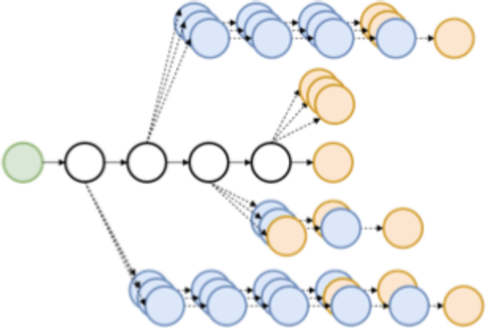
Applications:

- S. Liu, Z. Zhu, N. Ye, S. Guadarrama, K. Murphy
- Improved Image Captioning via Policy Gradient optimization of SPIDEr
- 2016

Image Captioning



Sampling a caption:



$$\nabla_\theta V_\theta(s_0) \approx \sum_{t=1}^{T} \sum_{g_t} [\pi_\theta(g_t|s_t) \nabla_\theta \log \pi_\theta(g_t|s_t) \times (Q_\theta(s_t, g_t) - B_\phi(s_t))] \quad (7)$$

$$L_\phi = \sum_t E_{s_t} E_{g_t} (Q_\theta(s_t, g_t) - B_\phi(s_t))^2 \quad (8)$$

Algorithm 1: PG training algorithm

```
Input: D = {(x^n, y^n) : n = 1 : N} ;
Train π_θ(g_{1:T}|x) using MLE on D ;
Train B_φ using MC estimates of Q_θ on a small subset
  of D;
for each epoch do
    for example (x^n, y^n) do
        Generate sequence g_{1:T} ~ π_θ(·|x^n) ;
        for t = 1 : T do
            Compute Q(g_{1:t-1}, g_t) for g_t with K
              Monte Carlo rollouts, using (6);
            Compute estimated baseline B_φ(g_{1:t-1});
        Compute G_θ = ∇_θ V_θ(s_0) using (7);
        Compute G_φ = ∇_φ L_φ;
        SGD update of θ using G_θ;
        SGD update of φ using G_φ;
```

Quiz:

- Why Policy Gradient?
- Techniques to improve vanilla Policy Gradient?

Important topics of this lecture

- Getting a feeling for reinforcement learning
- Understanding how to use reinforcement learning

What's next:

- Applied Machine Learning: CS 498
- Artificial Intelligence: ECE 440/CS 440
- MDPs, Reinforcement Learning: ECE 586
- Learning Algorithms: ECE 598 PV
- Machine Learning in Silicon: ECE 598
- Computational Inference and Learning: ECE 566
- Statistical Learning Theory: ECE 543/CS 598
- Computer Vision: ECE 549
- Digital Imaging: ECE 558