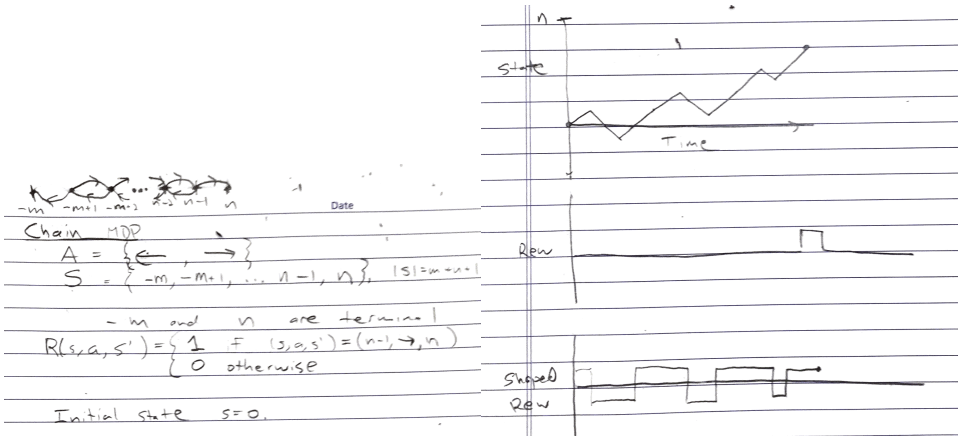
Variance Reduction for Policy Gradient Methods

March 13, 2017

Reward Shaping

Reward Shaping



Reward Shaping

- Reward shaping: $\tilde{r}(s, a, s') = r(s, a, s') + \gamma\Phi(s') - \Phi(s)$ for arbitrary "potential" Φ
- Theorem: $\tilde{r}$ admits the same optimal policies as r.[1]
  - Proof sketch: suppose Q^* satisfies Bellman equation ($\mathcal{T}Q = Q$). If we transform $r \to \tilde{r}$, policy's value function satisfies $\tilde{Q}(s, a) = Q^*(s, a) - \Phi(s)$
  - Q^* satisfies Bellman equation $\Rightarrow$ $\tilde{Q}$ also satisfies Bellman equation

[1] A. Y. Ng, D. Harada, and S. Russell. "Policy invariance under reward transformations: Theory and application to reward shaping". *ICML*. 1999.

Reward Shaping

- Theorem: $\tilde{R}$ admits the same optimal policies as R. A. Y. Ng, D. Harada, and S. Russell. "Policy invariance under reward transformations: Theory and application to reward shaping". *ICML*. 1999
- Alternative proof: advantage function is invariant. Let's look at effect on V^π and Q^π:

$$\begin{aligned}
&\mathbb{E}\left[r_0 + \gamma r_1 + \gamma^2 r_2 + \dots\right] && \text{condition on either } s_0 \text{ or } (s_0, a_0) \\
&= \mathbb{E}\left[\tilde{r}_0 + \gamma \tilde{r}_1 + \gamma^2 \tilde{r}_2 + \dots\right] \\
&= \mathbb{E}\left[(r_0 + \gamma\Phi(s_1) - \Phi(s_0)) + \gamma(r_1 + \gamma\Phi(s_2) - \Phi(s_1)) + \gamma^2(r_2 + \gamma\Phi(s_3) - \Phi(s_2)) + \dots\right] \\
&= \mathbb{E}\left[r_0 + \gamma r_1 + \gamma^2 r_2 + \dots - \Phi(s_0)\right]
\end{aligned}$$

Thus,

$$\begin{aligned}
\tilde{V}^\pi(s) &= V^\pi(s) - \Phi(s) \\
\tilde{Q}^\pi(s) &= Q^\pi(s, a) - \Phi(s) \\
\tilde{A}^\pi(s) &= A^\pi(s, a)
\end{aligned}$$

$A^\pi(s, \pi(s)) = 0$ at all states $\Rightarrow \pi$ is optimal

Reward Shaping and Problem Difficulty

- Shape with $\Phi = V^*$ $\Rightarrow$ problem is solved in one step of value iteration
- Shaping leaves policy gradient invariant (and just adds baseline to estimator)

$$
\begin{aligned}
&\mathbb{E}\big[\nabla_\theta \log \pi_\theta(a_0 \mid s_0)(r_0 + \gamma\Phi(s_1) - \Phi(s_0)) + \gamma(r_1 + \gamma\Phi(s_2) - \Phi(s_1)) \\
&\qquad + \gamma^2(r_2 + \gamma\Phi(s_3) - \Phi(s_2)) + \dots\big] \\
&= \mathbb{E}\left[\nabla_\theta \log \pi_\theta(a_0 \mid s_0)(r_0 + \gamma r_1 + \gamma^2 r_2 + \cdots - \Phi(s_0))\right] \\
&= \mathbb{E}\left[\nabla_\theta \log \pi_\theta(a_0 \mid s_0)(r_0 + \gamma r_1 + \gamma^2 r_2 + \dots)\right]
\end{aligned}
$$

Reward Shaping and Policy Gradients

- First note connection between shaped reward and advantage function:

$$\mathbb{E}_{s_1}\left[r_0 + \gamma V^\pi(s_1) - V^\pi(s_0) \mid s_0 = s, a_0 = a\right] = A^\pi(s,a)$$

Now considering the policy gradient and ignoring all but first shaped reward (i.e., pretend $\gamma = 0$ after shaping) we get

$$\mathbb{E}\left[\sum_t \nabla_\theta \log \pi_\theta(a_t \mid s_t)\tilde{r}_t\right] = \mathbb{E}\left[\sum_t \nabla_\theta \log \pi_\theta(a_t \mid s_t)(r_t + \gamma V^\pi(s_{t+1}) - V^\pi(s_t))\right]$$

$$= \mathbb{E}\left[\sum_t \nabla \log \pi_\theta(a_t \mid s_t) A^\pi(s_t, a_t)\right]$$

- Compromise: use more aggressive discount $\gamma\lambda$, with $\lambda \in (0,1)$: called generalized advantage estimation

$$\mathbb{E}\left[\sum_t \nabla_\theta \log \pi_\theta(a_t \mid s_t) \sum_{k=0}^{\infty} (\gamma\lambda)^k \tilde{r}_{t+k}\right]$$

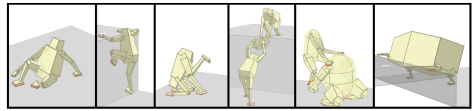
Reward Shaping – Summary

- Reward shaping transformation leaves policy gradient and optimal policy invariant
- Shaping with $\Phi \approx V^{\pi}$ makes consequences of actions more immediate
- Shaping, and then ignoring all but first term, gives policy gradient

[Aside] Reward Shaping: Very Important in Practice

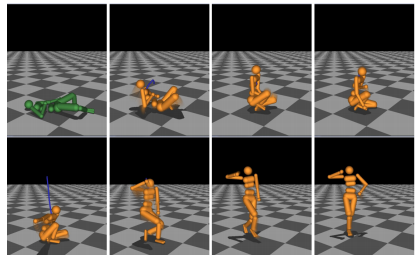
- I. Mordatch, E. Todorov, and Z. Popović. "Discovery of complex behaviors through contact-invariant optimization". *ACM Transactions on Graphics (TOG)* 31.4 (2012), p. 43

$$L(\mathbf{s}) = L_{\text{CI}}(\mathbf{s}) + L_{\text{Physics}}(\mathbf{s}) + L_{\text{Task}}(\mathbf{s}) + L_{\text{Hint}}(\mathbf{s})$$



- Y. Tassa, T. Erez, and E. Todorov. "Synthesis and stabilization of complex behaviors through online trajectory optimization". *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE. 2012, pp. 4906–4913

The state-cost is composed of 4 terms. The first term penalizes the horizontal distance (in the xy-plane) between the center-of-mass (CoM) and the mean of the feet positions. The second term penalizes the horizontal distance between the torso and the CoM. The third penalizes the vertical distance between the torso and a point 1.3m over the mean of the feet. All three terms use the smooth-abs norm (Figure 2).



Variance Reduction for Policy Gradients

Variance Reduction

- We have the following policy gradient formula:

$$\nabla_\theta \mathbb{E}_\tau [R] = \mathbb{E}_\tau \left[\sum_{t=0}^{T-1} \nabla_\theta \log \pi(a_t \mid s_t, \theta) A^\pi(s_t, a_t) \right]$$

- A^π is not known, but we can plug in $\hat{A}_t$, an *advantage estimator*
- Previously, we showed that taking

$$\hat{A}_t = r_t + r_{t+1} + r_{t+2} + \cdots - b(s_t)$$

for any function $b(s_t)$, gives an unbiased policy gradient estimator. $b(s_t) \approx V^\pi(s_t)$ gives variance reduction.

The Delayed Reward Problem

- With policy gradient methods, we are confounding the effect of multiple actions:

$$\hat{A}_t = r_t + r_{t+1} + r_{t+2} + \cdots - b(s_t)$$

  mixes effect of $a_t, a_{t+1}, a_{t+2}, \ldots$
- SNR of $\hat{A}_t$ scales roughly as $1/T$
  - Only a_t contributes to *signal* $A^\pi(s_t, a_t)$, but $a_{t+1}, a_{t+2}, \ldots$ contribute to noise.

Variance Reduction with Discounts

- Discount factor γ, $0 < \gamma < 1$, downweights the effect of rewards that are far in the future—ignore long term dependencies
- We can form an advantage estimator using the *discounted return*:

$$\hat{A}_t^{\gamma} = \underbrace{r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \dots}_{\text{discounted return}} - b(s_t)$$

  reduces to our previous estimator when $\gamma = 1$.
- So advantage has expectation zero, we should fit baseline to be *discounted value function*

$$V^{\pi,\gamma}(s) = \mathbb{E}_{\tau}\left[r_0 + \gamma r_1 + \gamma^2 r_2 + \dots \mid s_0 = s\right]$$

- Discount γ is similar to using a horizon of $1/(1-\gamma)$ timesteps
- $\hat{A}_t^{\gamma}$ is a biased estimator of the advantage function

Value Functions in the Future

- Baseline accounts for and removes the effect of *past* actions
- Can also use the value function to estimate future rewards

$$
\begin{aligned}
&r_t + \gamma V(s_{t+1}) && \text{cut off at one timestep} \\
&r_t + \gamma r_{t+1} + \gamma^2 V(s_{t+2}) && \text{cut off at two timesteps} \\
&\dots \\
&r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \dots && \infty \text{ timesteps (no } V\text{)}
\end{aligned}
$$

Value Functions in the Future

- Subtracting out baselines, we get advantage estimators

$$\hat{A}_t^{(1)} = r_t + \gamma V(s_{t+1}) - V(s_t)$$
$$\hat{A}_t^{(2)} = r_t + r_{t+1} + \gamma^2 V(s_{t+2}) - V(s_t)$$
$$\dots$$
$$\hat{A}_t^{(\infty)} = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \dots - V(s_t)$$

- $\hat{A}_t^{(1)}$ has low variance but high bias, $\hat{A}_t^{(\infty)}$ has high variance but low bias.
- Using intermediate k (say, 20) gives an intermediate amount of bias and variance

Finite-Horizon Methods: Advantage Actor-Critic

- A2C / A3C uses this fixed-horizon advantage estimator
- Pseudocode

for iteration$=1, 2, \ldots$ **do**
Agent acts for T timesteps (e.g., $T = 20$),
For each timestep t, compute

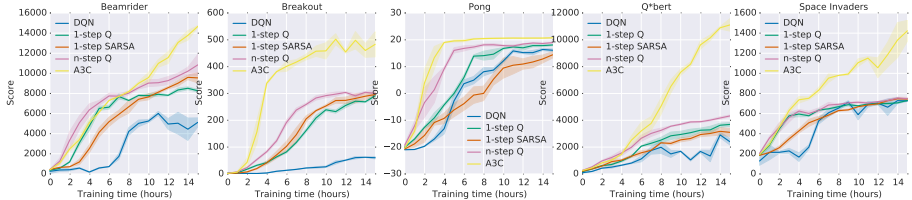
$$\hat{R}_t = r_t + \gamma r_{t+1} + \cdots + \gamma^{T-t+1} r_{T-1} + \gamma^{T-t} V(s_t)$$
$$\hat{A}_t = \hat{R}_t - V(s_t)$$

$\hat{R}_t$ is target value function, in regression problem
$\hat{A}_t$ is estimated advantage function
Compute loss gradient $g = \nabla_\theta \sum_{t=1}^{T} \left[-\log \pi_\theta(a_t \mid s_t) \hat{A}_t + c(V(s) - \hat{R}_t)^2 \right]$
g is plugged into a stochastic gradient descent variant, e.g., Adam.
end for

V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. P. Lillicrap, et al. "Asynchronous methods for deep reinforcement learning". (2016)

A3C Video

A3C Results


Beamrider
Breakout
Pong
Q*bert
Space Invaders
DQN
1-step Q
1-step SARSA
n-step Q
A3C
Score
Training time (hours)

Reward Shaping

TD(λ) Methods: Generalized Advantage Estimation

- Recall, finite-horizon advantage estimators

$$\hat{A}_t^{(k)} = r_t + \gamma r_{t+1} + \cdots + \gamma^{k-1} r_{t+k-1} + \gamma^k V(s_{t+k}) - V(s_t)$$

- Define the TD error $\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$
- By a telescoping sum,

$$\hat{A}_t^{(k)} = \delta_t + \gamma \delta_{t+1} + \cdots + \gamma^{k-1} \delta_{t+k-1}$$

- Take exponentially weighted average of finite-horizon estimators:

$$\hat{A}^\lambda = \hat{A}_t^{(1)} + \lambda \hat{A}_t^{(2)} + \lambda^2 \hat{A}_t^{(3)} + \ldots$$

- We obtain

$$\hat{A}_t^\lambda = \delta_t + (\gamma\lambda)\delta_{t+1} + (\gamma\lambda)^2 \delta_{t+2} + \ldots$$

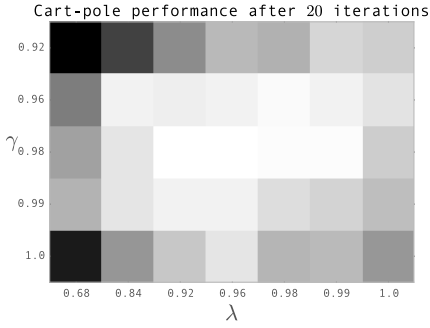
- This scheme named *generalized advantage estimation* (GAE) in [1], though versions have appeared earlier, e.g., [2]. Related to TD(λ)

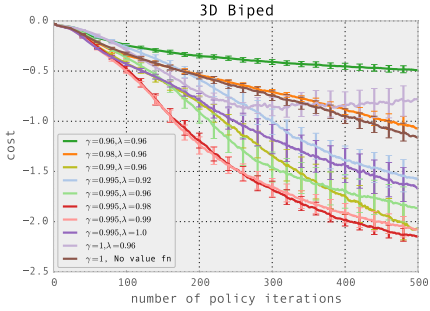
J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel. "High-dimensional continuous control using generalized advantage estimation". (2015)

H. Kimura and S. Kobayashi. "An Analysis of Actor/Critic Algorithms Using Eligibility Traces: Reinforcement Learning with Imperfect Value

Choosing parameters γ, λ

Performance as γ, λ are varied





TRPO+GAE Video