From Policy Gradient to Actor-Critic methods
The policy gradient derivation (3/3)

Olivier Sigaud
Sorbonne Université
http://people.isir.upmc.fr/sigaud
From Policy Gradient to Actor-Critic methods

Policy Gradient with Baseline

Reminder:

\[ \nabla_\theta J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \sum_{t=1}^{H} \nabla_\theta \log \pi_\theta(a_t^{(i)} | s_t^{(i)}) \left[ \sum_{k=t}^{H} \gamma^k r(s_k^{(i)}, a_k^{(i)}) \right] \]  

If all rewards are positive, the gradient increases all probabilities

But with renormalization, only the largest increases emerge

We can subtract a “baseline” to (1) without changing its mean:

\[ \nabla_\theta J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \sum_{t=1}^{H} \nabla_\theta \log \pi_\theta(a_t^{(i)} | s_t^{(i)}) \left[ \sum_{k=t}^{H} \gamma^k r(s_k^{(i)}, a_k^{(i)}) - b \right] \]

A first baseline is the average return \( \bar{r} \) over all states of the batch

Intuition: returns greater than average get positive, smaller get negative

Use \( (r_t^{(i)} - \bar{r}) \) and divide by std \( \rightarrow \) get a mean = 0 and a std = 1

This improves variance (does the job of renormalization)

Suggested in https://www.youtube.com/watch?v=tqrcjHuNdmQ
Algorithm 4: adding a constant baseline

\[ \sum_{t=1}^{H} \gamma^{t-1}(r_t - \bar{r})/std(r) \]

- Estimate \( \bar{r} \) and \( std(r) \) from all rollouts
- Same as Algorithm 2, using \( (r^{(i)}_t - \bar{r})/std(r) \)
- Suffers from even less variance
- Does not work if all rewards \( r \) are identical (e.g. CartPole)
Policy Gradient with state-dependent baseline

- No impact on the gradient as long as the baseline does not depend on action
- A better baseline is \( b(s_t) = V^\pi(s_t) = \mathbb{E}_\tau[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \ldots + \gamma^{H-t} r_H] \)
- The expectation can be approximated from the batch of trajectories
- Thus we get

\[
\nabla_\theta J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \sum_{t=1}^{H} \nabla_\theta \log \pi_\theta(a_{(i)t}\mid s_{(i)t}) [Q^\pi_\theta(s_{(i)t}\mid a_{(i)t}) - V^\pi_\theta(s_{(i)t})]
\]

- \( A^\pi(s_t, a_t) = Q^\pi(s_t\mid a_t) - V^\pi(s_t) \) is the advantage function
- And we get

\[
\nabla_\theta J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \sum_{t=1}^{H} \nabla_\theta \log \pi_\theta(a_{(i)t}\mid s_{(i)t}) A^\pi_\theta(s_{(i)t}, a_{(i)t})
\]

https://www.youtube.com/watch?v=S_gwYj1Q-44 (27')

From Policy Gradient to Actor-Critic methods

Policy Gradient with Baseline

Estimating $V^\pi(s)$

- As for estimating $Q^\pi(s, a)$, but simpler
- Two approaches:
  - Monte Carlo estimate: Regression against empirical return
    $$\phi_{j+1} \rightarrow \arg \min_{\phi_j} \frac{1}{m} \sum_{i=1}^{m} \sum_{t=1}^{H} \left( \sum_{k=t}^{H} r(s_{t}^{(i)}, a_{t}^{(i)}) - \hat{V}_{\phi_j}^{\pi}(s_{t}^{(i)}) \right)^2$$
  - Temporal Difference estimate: init $\hat{V}_{\phi_0}^{\pi}$ and fit using $(s, a, r, s')$ data
    $$\phi_{j+1} \rightarrow \min_{\phi_j} \sum_{(s, a, r, s')} \|r + \gamma \hat{V}_{\phi_j}^{\pi}(s') - \hat{V}_{\phi_j}^{\pi}(s)\|^2$$
- May need some regularization to prevent large steps in $\phi$
From Policy Gradient to Actor-Critic methods

Policy Gradient with Baseline

Algorithm 5: adding a state-dependent baseline

- Learn $\hat{V}_\phi^\pi$ from TD, from MC rollouts, or compute $V^\pi_\theta(s^{(i)}_t)$ from MC
- Learn $\hat{Q}_\phi^\pi$ from TD, from MC rollouts, or compute $Q^\pi_\theta(s^{(i)}_t, a^{(i)}_t)$ from MC
- Compute $A^\pi(s^{(i)}_t | a^{(i)}_t) = Q^\pi_\phi(s^{(i)}_t, a^{(i)}_t) - \hat{V}_\phi^\pi(s^{(i)}_t)$
- Or even learn $\hat{A}_\phi^\pi$ directly from TD updates using $A^\pi(s_t, a_t) = \mathbb{E}[\delta_t]$
- Same as Algorithm 3 using $A^\pi_\theta(s^{(i)}_t | a^{(i)}_t)$ instead of $Q^\pi_\theta(s^{(i)}_t | a^{(i)}_t)$
- Suffers from even less variance
From Policy Gradient to Actor-Critic methods

Policy Gradient with Baseline

Synthesis

\[ \nabla_{\theta} J(\theta) = \mathbb{E}_{s_t, a_t} \pi_{\theta}(.) [\nabla_{\theta} \log \pi_{\theta}(a_t(s_t))^i | s_t] ] \psi_t \] where \( \psi_t \) can be:

1. \( \sum_{t=0}^{H} \gamma^t r_t \): total (discounted) reward of trajectory
2. \( \sum_{k=t}^{H} \gamma^{k-t} r_k \): sum of rewards after \( a_t \)
3. \( \sum_{k=t}^{H} \gamma^{k-t} r_k - b(s_t) \): sum of rewards after \( a_t \) with baseline
4. \( \delta_t = r_t + \gamma V^\pi(s_{t+1}) - V^\pi(s_t) \): TD error, with \( V^\pi(s_t) = \mathbb{E}_{a_t} [\sum_{k=0}^{H} \gamma^k r_{t+k}] \)
5. \( \hat{Q}_\phi^\pi(s_t, a_t) = \mathbb{E}_{a_{t+1}} [\sum_{k=0}^{H} \gamma^k r_{t+k+l}] \): action-value function
6. \( \hat{A}_\phi^\pi(s_t, a_t) = \hat{Q}_\phi^\pi(s_t, a_t) - \hat{V}_\phi^\pi(s_t) = \mathbb{E}[\delta_t] \), advantage function

▶ Next lesson: Difference to Actor-Critic

Any question?

Send mail to: Olivier.Sigaud@upmc.fr
High-dimensional continuous control using generalized advantage estimation.

Ronald J. Williams.
Simple statistical gradient-following algorithms for connectionist reinforcement learning.
ISSN 0885-6125.