Comparing Policy-Gradient Algorithms

Lucas Berry

Comp 767

April 7th, 2017

Policy Gradient Methods

- Action-Value methods, when used simultaneously with linear function approximation, have been shown not to converge. (Q-Learning Diverges)
- This fueled the search for other methods with better convergence properties.

Policy Gradient Methods

- As opposed to action-value methods policy gradient methods directly parameterize the policy, θ.
- θ is adjusted according to the gradient of the overall performance:

$$\Delta\theta = \alpha\hat{\nabla}_\theta J(\theta).$$

Advantages

- The existence of the gradient $\nabla_\theta J(\theta)$. Action-value gradients are either zero or undefined.
- One can capture and find stochastic policies.
- Policy gradient methods do not need to find a maximum. This can be problematic in a large list.
- Can handle continuous action spaces as well.

Two ways of formulating objective

1. Average Reward Formulation:

$$J(\pi) = \lim_{n->\infty} \frac{1}{n} E\{r_1 + r_2 + ... + r_n | \pi\}$$

2. Discounted Return (designated start state):

$$J(\pi) = E_\pi \left(\sum_{t=0}^{\infty} \gamma^{k-1} r_k \right)$$

Gradient

One can show the gradient to be,

$$\nabla_\theta J = \sum_s d^\pi(s) \sum_a Q^\pi(s,a) \nabla_\theta \pi(s,a)$$

where d^π is the expected number times you reach state s. The proof is contained in our book page 269.

Update

The update can then be written as:

$$\theta_{t+1} = \theta_t + \alpha \frac{\hat{Q}^\pi(s_t, a_t)}{\pi(s_t, a_t)} \nabla_\theta \pi(s_t, a_t).$$

Now all that is left is to determine how to estimate $Q^\pi(s, a)$.

Estimating $Q^{\pi}(s, a)$

REINFORCE uses a Monte Carlo approach to derive an estimate. While Konda and Tsitsiklis (2000) and Sutton et al. (2000) propose a linear parameterize approximation,

$$q(s, a) = w^{T} \frac{\nabla_{\theta} \pi(s, a)}{\pi(s, a)},$$

where w is the parameter vector and

$$w = \arg\min_{w} \sum_{t} (R_t - q(s_t, a_t))^2.$$

This estimate is unbiased like that of the REINFORCE.

Unbiased doesn't always help

Unfortunately this approach does not outperform REINFORCE.

Theorem

For any batch of data D, the total update from the single-action algorithm is the same for both $\hat{Q}^\pi(s_t, a_t) = \hat{Q}^\pi_t$ and $\hat{Q}^\pi(s_t, a_t) = q(s_t, a_t)$. That is

$$\sum_{t \in D} \hat{Q}^\pi_t \frac{\nabla_\theta \pi(s_t, a_t)}{\pi(s_t, a_t)} = \sum_{t \in D} q(s_t, a_t) \frac{\nabla_\theta \pi(s_t, a_t)}{\pi(s_t, a_t)}.$$

Proof

Minimizing w for the squared error we need to take the gradient and set it equal to zero,

$$
\begin{aligned}
0 &= 2 \sum_{t \in D} \left(\hat{Q}_t^{\pi} - q(s_t, a_t) \right) \nabla_w q(s_t, a_t) \\
&= \sum_{t \in D} \left(\hat{Q}_t^{\pi} - q(s_t, a_t) \right) \frac{\nabla_\theta \pi(s_t, a_t)}{\pi(s_t, a_t)}.
\end{aligned}
$$

Rearranging the terms shows the theorem. Setting $\hat{Q}_t^{\pi}$ shows the relationship with REINFORCE.

Conclusion

Using the proposed unbiased parametrization performs no better than REINFORCE as it does nothing to reduce the variance. Thus in this case REINFORCE has it's advantages as it is simpler, it does not need to estimate w.

References

- Konda, V. R., Tsitsiklis, J. N. (2000) Actor-critic algorithms
- Sutton, R., McAllester, D., Singh, S., Mansour, Y. (2000). Policy gradient methods for reinforcement learning with function approximation.
- Sutton, R., Singh, S., McAllester D. Comparing Policy-Gradient Algorithms