Policy Gradient Methods - Actor Citic

Wonseok Jung

1. Actor-Critic Methods

- Reinforce with baseline 방법은 policy와 state-value를 모두 학습한다.
- 하지만 여기서 state-value가 critic이 아닌 baseline 이기에 actor-critic이라고 부르지 않는다.

1.1 Applying Bootstrap

- REINFORCE 방법에서는 BootStrap을 하지 않고 Terminal state 까지 받은 총 reward G_t 를 Baseline과의 차이를 계산한다.
- Bootstrapping으로 인해 생기는 bias는 variance를 낮추고 learning 속도를 빠르게 한다.
- 반면 REINFORCE는 unbias하지만, variance가 높으며 learning 속도가 느리다.
- 또한 REINFORCE 는 online으로 학습할수 없기에 continuous problem에 적합하지 않다.

1.2 REINFORCE algorithm with bootstrapping

- Bootstrapping의 장점을 이용하여 REINFORCE algorithm에 적용한 알고리즘을 Actor-Critic Methods 라고 한다.
- Actor-Critic Methods의 update Rule은 다음과 같다.

$$
\begin{aligned}
\boldsymbol{\theta}_{t+1} &\doteq \boldsymbol{\theta}_t + \alpha\Big(G_{t:t+1} - \hat{v}(S_t,\mathbf{w})\Big)\frac{\nabla\pi(A_t|S_t,\boldsymbol{\theta}_t)}{\pi(A_t|S_t,\boldsymbol{\theta}_t)} \\
&= \boldsymbol{\theta}_t + \alpha\Big(R_{t+1} + \gamma\hat{v}(S_{t+1},\mathbf{w}) - \hat{v}(S_t,\mathbf{w})\Big)\frac{\nabla\pi(A_t|S_t,\boldsymbol{\theta}_t)}{\pi(A_t|S_t,\boldsymbol{\theta}_t)} \\
&= \boldsymbol{\theta}_t + \alpha\delta_t\frac{\nabla\pi(A_t|S_t,\boldsymbol{\theta}_t)}{\pi(A_t|S_t,\boldsymbol{\theta}_t)}.
\end{aligned}
$$

One-step Actor-critic (Episodic)

One-step Actor–Critic (episodic), for estimating $\pi_{\boldsymbol{\theta}} \approx \pi_*$

Input: a differentiable policy parameterization $\pi(a|s,\boldsymbol{\theta})$
Input: a differentiable state-value function parameterization $\hat{v}(s,\mathbf{w})$
Parameters: step sizes $\alpha^{\boldsymbol{\theta}} > 0$, $\alpha^{\mathbf{w}} > 0$
Initialize policy parameter $\boldsymbol{\theta} \in \mathbb{R}^{d'}$ and state-value weights $\mathbf{w} \in \mathbb{R}^{d}$ (e.g., to $\mathbf{0}$)
Loop forever (for each episode):
  Initialize S (first state of episode)
  $I \leftarrow 1$
  Loop while S is not terminal (for each time step):
    $A \sim \pi(\cdot|S,\boldsymbol{\theta})$
    Take action A, observe S', R
    $\delta \leftarrow R + \gamma \hat{v}(S',\mathbf{w}) - \hat{v}(S,\mathbf{w})$ (if S' is terminal, then $\hat{v}(S',\mathbf{w}) \doteq 0$)
    $\mathbf{w} \leftarrow \mathbf{w} + \alpha^{\mathbf{w}} I \delta \nabla \hat{v}(S,\mathbf{w})$
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha^{\boldsymbol{\theta}} I \delta \nabla \ln \pi(A|S,\boldsymbol{\theta})$
    $I \leftarrow \gamma I$
    $S \leftarrow S'$

Actor-Critic with Eligibility Traces(episodic)

Actor–Critic with Eligibility Traces (episodic), for estimating $\pi_{\boldsymbol{\theta}} \approx \pi_*$

Input: a differentiable policy parameterization $\pi(a|s,\boldsymbol{\theta})$
Input: a differentiable state-value function parameterization $\hat{v}(s,\mathbf{w})$
Parameters: trace-decay rates $\lambda^{\boldsymbol{\theta}} \in [0,1]$, $\lambda^{\mathbf{w}} \in [0,1]$; step sizes $\alpha^{\boldsymbol{\theta}} > 0$, $\alpha^{\mathbf{w}} > 0$
Initialize policy parameter $\boldsymbol{\theta} \in \mathbb{R}^{d'}$ and state-value weights $\mathbf{w} \in \mathbb{R}^{d}$ (e.g., to $\mathbf{0}$)
Loop forever (for each episode):
  Initialize S (first state of episode)
  $\mathbf{z}^{\boldsymbol{\theta}} \leftarrow \mathbf{0}$ (d'-component eligibility trace vector)
  $\mathbf{z}^{\mathbf{w}} \leftarrow \mathbf{0}$ (d-component eligibility trace vector)
  $I \leftarrow 1$
  Loop while S is not terminal (for each time step):
    $A \sim \pi(\cdot|S,\boldsymbol{\theta})$
    Take action A, observe S', R
    $\delta \leftarrow R + \gamma \hat{v}(S',\mathbf{w}) - \hat{v}(S,\mathbf{w})$ (if S' is terminal, then $\hat{v}(S',\mathbf{w}) \doteq 0$)
    $\mathbf{z}^{\mathbf{w}} \leftarrow \gamma \lambda^{\mathbf{w}} \mathbf{z}^{\mathbf{w}} + I \nabla \hat{v}(S,\mathbf{w})$
    $\mathbf{z}^{\boldsymbol{\theta}} \leftarrow \gamma \lambda^{\boldsymbol{\theta}} \mathbf{z}^{\boldsymbol{\theta}} + I \nabla \ln \pi(A|S,\boldsymbol{\theta})$
    $\mathbf{w} \leftarrow \mathbf{w} + \alpha^{\mathbf{w}} \delta \mathbf{z}^{\mathbf{w}}$
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha^{\boldsymbol{\theta}} \delta \mathbf{z}^{\boldsymbol{\theta}}$
    $I \leftarrow \gamma I$
    $S \leftarrow S'$

Summary

- Q-leraning, SARSA, MC와 같은 알고리즘은 action value를 측정하고, 이를 사용하여 actoin 을 선택한다.
- 여기서는 action value를 estimate하지 않아도 parameterized policy를 배워 action을 선택하는 방법을 알아보았다.
- 이를 Policy gradient 방법이라고 한다.
- Policy gradient는 각 action을 선택할 확률을 구할 수 있으며 더 이상 $\epsilon - greedy$와 같은 exploration 방법은 사용하지 않는다.

- REINFORCE methods 는 state-value function을 baseline으로 추가하여 variance를 줄인다.
- Bootstrapping을 사용한 TD방법은 Monte Carlo보다 variance를 줄이는 효과가 있다.
- 이 방법은 REINFORCE 알고리즘에 적용하여 policy에 의해 선택된 action을 critic하는 algorithm을 Actor-Critic 이라고 한다.

References

Policy Gradient 개념 및 REINFORCE 알고리즘, REINFORCE 알고리즘 with baseline 설명

https://github.com/wonseokjung/ReinforcementLearning_byWonseok/blob/master/8. Policy Gradient Methods/1.PG_REINFORCE/pgtoReinbase.pdf