CMPUT 365: Review

Rupam Mahmood

April 8, 2022

Little bit about policy gradient and actor critic methods

- We have been using policies based on values like epsilon greedy based on an action-value estimate
- how about we represent policies explicitly as a parameterized functions just like how represented value estimates under function approximation
- But we do so directly without a reference to the value estimate

agent wants to maximize the reward aggregate.

> selection. We use the notation $\boldsymbol{\theta} \in \mathbb{R}^{d'}$ for the policy's parameter vector. Thus we write $\pi(a|s, \boldsymbol{\theta}) = \Pr\{A_t = a \mid S_t = s, \boldsymbol{\theta}_t = \boldsymbol{\theta}\}$ for the probability that action a is taken at time t given that the environment is in state s at time t with parameter $\boldsymbol{\theta}$. If a method uses a

- We already learned about NNs. So, pi could be represented by an NN. It will take S (state) as an input and output parameters of a distribution of actions; for example, theta can be mean mu and standard deviation sigma of a normal distribution. Then the action will be drawn from N(mu, sigma^2)

depend on the state !! ∴ μ(S), σ(S).

- And we already learned about stochastic gradient descent; we can use the same mechanism to maximize the objective function of the agent directly

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha \widehat{\nabla J(\boldsymbol{\theta}_t)},$$

What will be the objective?

- Following is a sensible objective because certainly we want to maximize the value from the start state $J(\boldsymbol{\theta}) \doteq v_{\pi_{\boldsymbol{\theta}}}(s_0)$
- And this objective is a function of the policy parameters theta (practically neural network weights) because the value function is dependent on the policy
- How do we perform stochastic-gradient update on an objective?
- From supervised learning & SGD notes, we learned that we can find an unbiased estimate of the objective as a sample loss, for example, for MSE we can form the squared prediction error as a single sample loss
- Then we can take a gradient of that sample loss
- Alternatively, we can find an unbiased estimate of the gradient of the true objective
- Policy gradient methods does the latter

Forming an unbiased estimate of the true gradient

$$\nabla J(\boldsymbol{\theta}) = \nabla v_\pi(s_0) \propto \sum_s \mu(s) \sum_a \nabla \pi(a|s) q_\pi(s,a)$$

$$= E_{S_t \sim \mu, A \sim \pi} \left[G_t \nabla \log \pi(A_t|S_t) \right]$$

According to the policy gradient theorem

In the above we wrote the gradient of the objective J (which we can also call the true gradient) as an expectation of a random variable

Then the random variable becomes an unbiased estimate of the true gradient

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha \widehat{\nabla J(\boldsymbol{\theta}_t)}$$

$$\widehat{\nabla J(\boldsymbol{\theta})} = G_t \nabla \log \pi_{\boldsymbol{\theta}}(A_t|S_t)$$

Then we can use the estimate to make stochastic-gradient updates like above
This method is known as Reinforce

Actor-critic methods

Just like TD forms a biased estimate, actor-critic forms a biased estimate too, by applying the same principle of bootstrapping on the Reinforce method

Reinforce: $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha \widehat{\nabla J(\boldsymbol{\theta}_t)}$, $\quad \widehat{\nabla J(\boldsymbol{\theta})} = G_t \nabla \log \pi_{\boldsymbol{\theta}}(A_t|S_t)$

Actor-critic:

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

policy Gradient 방법의 특징 :
- don't need to compute the value estimate to draw the actions. (don't need to compute the value of every action)
- You can use parameter policies

you need to estimate V

⇒ update every time step !! ∴ 학습) TD. (bootstraping).

Here, we subtracted the return by a value estimate, which is surprisingly still unbiased!
But then replaced the full return with a one-step return based on the current value estimate
And we get our beloved TD error back!
This method requires estimating the value function, which is known as the critic here
Pi is the actor

DR

wait sorry I think im wrong when I saw ppo can update per step, since it requires the computation of the advantage function, doesn't that mean that it needs to see the entire episode to compute discounted rewards?

...

say*

Gautham Vasan对所有人说 下午1:25

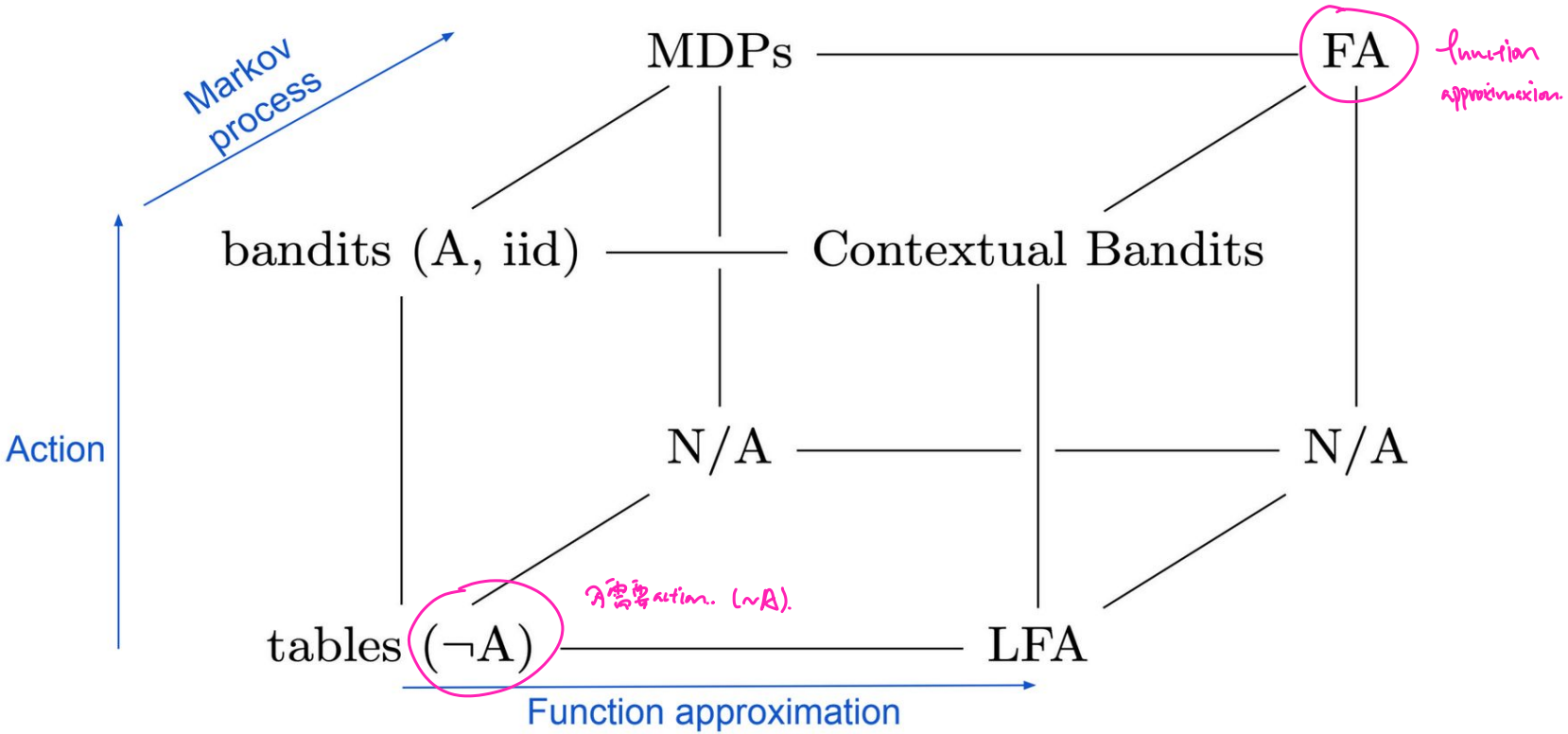
@Deepak PPO does not directly use G_t, we use an advantage estimate based on the value estimate of the state instead.

Some of the things we can do based on this course

- The paper for the robot demo we saw during introduction is now accepted for publication (ICRA 2022) and released with open-source code

- Ongoing work: the demo clearly shows the difference between initial and later performance of iterative methods

We learned about the elements necessary for building and understanding general-purpose learning mechanisms



We learned some life lessons too

- Seek simplicity
- Formulate the problem really well before starting to solve it!
- Seek generality
- Be robust by relying on less external things
- Build foundations internally!