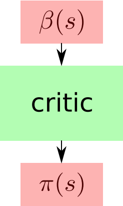
From Policy Gradient to Actor-Critic methods

On-policy versus Off-policy

Olivier Sigaud

Sorbonne Université
http://people.isir.upmc.fr/sigaud

Basic concepts



- To understand the distinction, one must consider three objects:
  - The behavior policy $\beta(s)$ used to generate samples.
  - The critic, which is generally $V(s)$ or $Q(s, a)$
  - The target policy $\pi(s)$ used to control the system in exploitation mode.

Singh, S. P., Jaakkola, T., Littman, M. L., & Szepesvári, C. (2000) Convergence results for single-step on-policy reinforcement-learning algorithms. *Machine learning*, 38(3):287–308

Off-policy learning: definitions

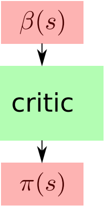
- "Off-policy learning": learning about one way of behaving, called the *target policy*, from data generated by another way of selecting actions, called the *behavior policy*.
- "Off-policy data": training samples which were not generated using $\pi(s)$
- Two research topics:
  - Off-policy policy evaluation (not covered): how can we get the critic related to a policy given off-policy data?
  - Off-policy control: how can we get an optimal policy by training a policy given off-policy data?
- Ex: stochastic behavior policy, deterministic target policy.
- Training data can be more or less off-policy (close to data from $\pi(s)$)
- An algo. is said off-policy if it reaches the optimal policy using off-policy data.

Maei, H. R., Szepesvári, C., Bhatnagar, S., & Sutton, R. S. (2010) Toward off-policy learning control with function approximation. *ICML*, pages 719–726.

Why preferring off-policy to on-policy control?

- Reusing old data, e.g. from a replay buffer (sample efficiency)
- More freedom for exploration
- Learning from human data (imitation)
- Transfer between policies in a multitask context

An illustrative study: two steps



- Open-loop study
  - Use uniform sampling as "behavior policy" (few assumptions)
  - No exploration issue, no bias towards good samples
  - NB: in uniform sampling, samples do not correspond to an agent trajectory
  - Study critic learning from these samples
- Then close the loop:
  - Use the target policy + some exploration as behavior policy
  - If the target policy gets good, bias more towards good samples

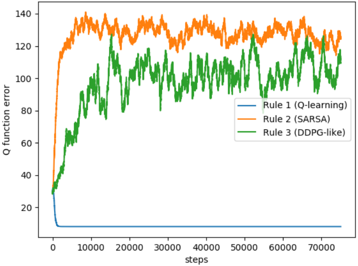
Learning a critic from samples

- General format of samples S: $(\mathbf{s}_t, \mathbf{a}_t, \mathbf{r}_t, \mathbf{s}_{t+1}, \mathbf{a}')$
- Makes it possible to apply a general update rule:

$$Q(\mathbf{s}_t, \mathbf{a}_t) \leftarrow Q(\mathbf{s}_t, \mathbf{a}_t) + \alpha[\mathbf{r}_t + \gamma Q(\mathbf{s}_{t+1}, \mathbf{a}') - Q(\mathbf{s}_t, \mathbf{a}_t)]$$

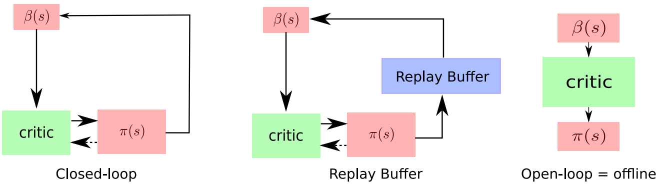
- There are three possible update rules:
  1. $a' = \mathrm{argmax}\, a Q(\mathbf{s}_{t+1}, \mathbf{a})$ (corresponds to Q-LEARNING)
  2. $a' = \beta(\mathbf{s}_{t+1})$ (corresponds to SARSA)
  3. $a' = \pi(\mathbf{s}_{t+1})$ (corresponds e.g. to DDPG, an ACTOR-CRITIC algorithm)

Results



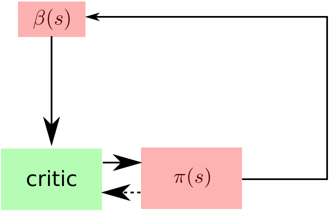
- Rule 1 learns an optimal critic (thus Q-LEARNING is truly off-policy)
- Rule 2 fails (thus SARSA is not off-policy)
- Rule 3 fails too (thus an algorithm like DDPG is not truly off-policy!)
- NB: different ACTOR-CRITIC implementations behave differently
- E.g. if the critic estimates $V(\mathrm{s})$, then equivalent to Rule 1

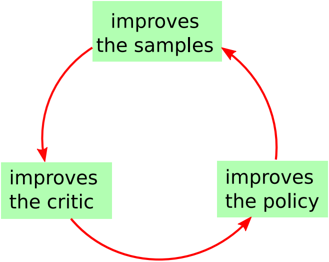
Three contexts



- Closed-loop case: data is on-policy
- Replay Buffer (RB) case: intermediate
- Open-loop case: offline RL

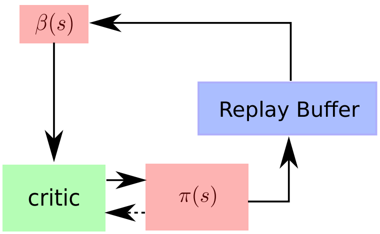
Closing the loop





- If $\beta(\mathbf{s}) = \pi^*(\mathbf{s})$, then Rules 2 and 3 are equivalent,
- Furthermore, $Q(\mathbf{s}, \mathbf{a})$ will converge to $Q^*(\mathbf{s}, \mathbf{a})$, and Rule 1 will be equivalent too.
- Quite obviously, Q-LEARNING still works
- SARSA and ACTOR-CRITIC work too: $\beta(\mathbf{s})$ becomes "Greedy in the Limit of Infinite Exploration" (GLIE)
- In the closed-loop case, data is on-policy, on-policy algorithms can converge too.
- An on-policy algorithm can only converge if the data is on-policy.

Replay buffer case

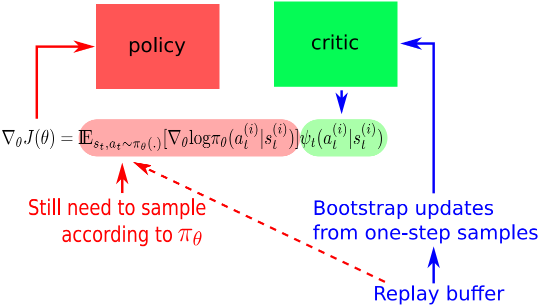


- With a replay buffer, $\beta(\mathbf{s})$ is generally close enough to $\pi(\mathbf{s})$
- The bigger the RB, the more off-policy the data
- Being (at least partly) off-policy is a necessary condition for using a replay buffer

Off-policy and actor-critic

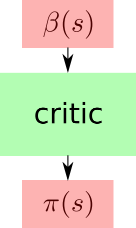
- Because AC algorithms use a TD mechanism, they perform one-step updates
- Performing one-step updates is a necessary condition for using a replay buffer
- Thus AC algos often use a replay buffer (A2C and A3C are counter-examples)
- Thus AC algos are often said off-policy
- DDPG, TD3 and SAC are AC algos, they use a replay buffer and they are said off-policy

Off-policy RB algorithms: remark



- DDPG, TD3 and SAC use off-policy samples to update the critic
- To udpate the actor, they use $\delta_t = r_t + \gamma \hat{Q}_{\boldsymbol{\phi}}^{\pi_{\boldsymbol{\theta}}}(\mathbf{s}_{t+1}, \pi_{\boldsymbol{\theta}}(\mathbf{s}_{t+1})) - \hat{Q}_{\boldsymbol{\phi}}^{\pi_{\boldsymbol{\theta}}}(\mathbf{s}_t, \mathbf{a}_t)$
- Thus updating the actor uses on-policy samples
- Alternative: $\delta_t = r_t + \gamma \hat{Q}_{\boldsymbol{\phi}}^{\pi_{\boldsymbol{\theta}}}(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}) - \hat{Q}_{\boldsymbol{\phi}}^{\pi_{\boldsymbol{\theta}}}(\mathbf{s}_t, \mathbf{a}_t)$
- Using samples $(\mathbf{s}_t, \mathbf{a}_t, \mathbf{r}_t, \mathbf{s}_{t+1}, \mathbf{a}_{t+1})$
- Would be a deep SARSA

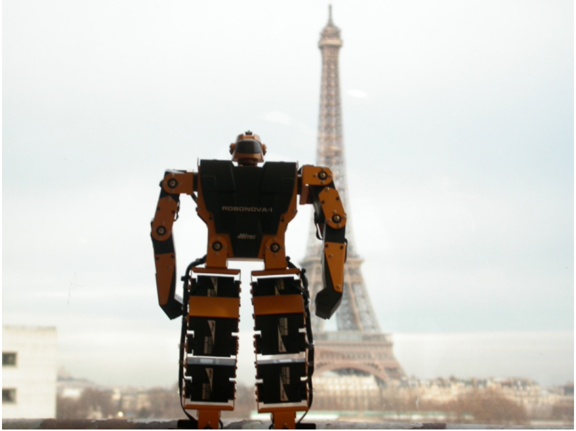
Offline RL case



- Q-LEARNING is the only truly off-policy algorithm that I know about
- Offline RL: find the assumptions on the data so as to guarantee the optimal behavior can be found

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020

Any question?



Send mail to: Olivier.Sigaud@upmc.fr

Levine, S., Kumar, A., Tucker, G., and Fu, J. (2020).
Offline reinforcement learning: Tutorial, review, and perspectives on open problems.
arXiv preprint arXiv:2005.01643.

Maei, H. R., Szepesvári, C., Bhatnagar, S., and Sutton, R. S. (2010).
Toward off-policy learning control with function approximation.
In *ICML*, pages 719–726.

Singh, S. P., Jaakkola, T., Littman, M. L., and Szepesvári, C. (2000).
Convergence results for single-step on-policy reinforcement-learning algorithms.
Machine learning, 38(3):287–308.