Imitation learning and Behavioral cloning

Olivier Sigaud

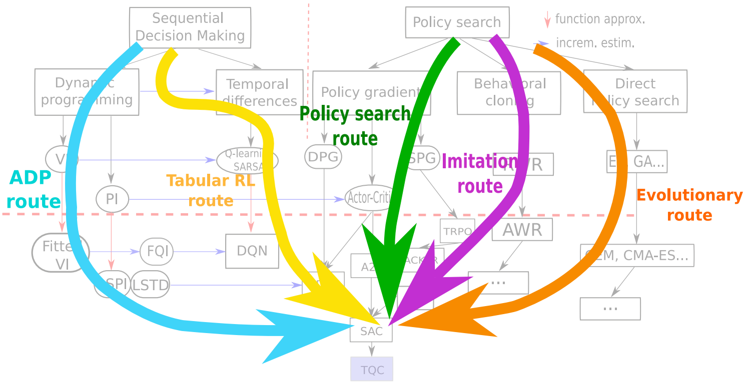
Sorbonne Université
http://people.isir.upmc.fr/sigaud

INSTITUT DES SYSTÈMES INTELLIGENTS ET DE ROBOTIQUE

Families of advanced RL mechanisms

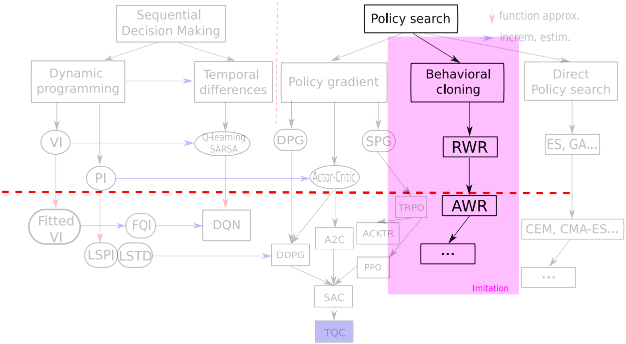
- Advanced exploration
- Imitation learning
- Goal-conditioned RL
- Hierarchical RL
- Learning from multiple agents
- Open-Ended Autotelic RL
- Human-in-the-loop RL (RLHP, RLHF...)
- Model-based RL
- Combinations of the above
- Hardly mentioned: Meta-RL, Offline RL
- Multi-Agent RL, Safe RL ...

The five routes to deep RL



- Reminder: five different ways to come to Deep RL

The Imitation learning route



- A very efficient route, with growing interest
- From imitation learning to offline RL

From policy gradient to off-policy learning

- Consider the policy search setting where you have a set of trajectories τ and the corresponding rewards $r(\tau)$
- In the policy gradient setting, you consider that you know the policy that generated these trajectories, and you write

$$P(\tau^{(i)}, \theta) = \prod_{t=1}^{H} p(s_{t+1}^{(i)} | s_t^{(i)}, a_t^{(i)}).\pi_\theta(a_t^{(i)} | s_t^{(i)})$$

- What if you do not know the policy? What can you do?
- This is the essence of the offline (and off-policy) setting

Learning a policy from regression

- An obvious thing to do is learn a policy from the trajectories using regression
- The learned policy should perform as the observed trajectories
- Provided rich enough trajectories, it should generalize to unseen states
- This is a form of learning from demonstration called behavioral cloning

Behavioral cloning

- Assume we have a set of expert trajectories,
- Data is a list of pairs $(\mathbf{s}_t^{(i)}, \mathbf{a}_t^{(i)})$, t is time, H is horizon, i is the trajectory index
- If the trajectories are optimal, behavioral cloning is a good option
- Use regression to find a policy $\pi_{\boldsymbol{\theta}}$ behaving as close as possible to data
- Use a validation set to avoid overfitting
- If the policy $\pi_{\boldsymbol{\theta}}$ is deterministic, this amounts to minimizing the loss function:

$$Loss(\boldsymbol{\theta}) = \frac{1}{m} \sum_{i=1}^{m} \sum_{t=1}^{H} (\mathbf{a}_t^{(i)} - \pi_{\boldsymbol{\theta}}(\mathbf{s}_t^{(i)}))^2$$

- If the policy $\pi_{\boldsymbol{\theta}}$ is stochastic, a standard approach (among many others) consists in minimizing the log likelihood loss function:

$$Loss(\boldsymbol{\theta}) = -\frac{1}{m} \sum_{i=1}^{m} \sum_{t=1}^{H} \log \pi_{\boldsymbol{\theta}}(\mathbf{a}_t^{(i)} | \mathbf{s}_t^{(i)})$$

- But the obtained policy does not perform better than observed trajectories
- Can we do better?

Reward Weighted Regression

- Now, if the expert trajectories are not optimal
- Let $R(\tau)$ be the return of trajectory τ
- Still use regression, but weight each sample depending on the return of the corresponding trajectory
- That is, imitate "more strongly" what is good in the batch than what is bad
- Still use a validation set to avoid overfitting
- If the policy $\pi_{\boldsymbol{\theta}}$ is deterministic, this amounts to minimizing the loss function:

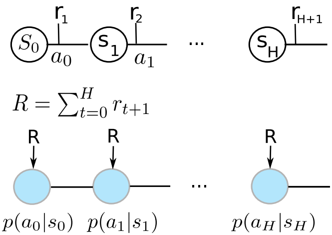
$$Loss(\boldsymbol{\theta}) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{t=1}^{H}(\mathbf{a}_t^{(i)} - \pi_{\boldsymbol{\theta}}(\mathbf{s}_t^{(i)}))^2 R(\tau^{(i)})$$

- If the policy $\pi_{\boldsymbol{\theta}}$ is stochastic, we minimize the function:

$$Loss(\boldsymbol{\theta}) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{t=1}^{H}\log\pi_{\boldsymbol{\theta}}(\mathbf{a}_t^{(i)}|\mathbf{s}_t^{(i)})R(\tau^{(i)}) \quad (1)$$

- Then we can iterate: generate new data from the new policy, and so on

Reminder: the most basic PG algorithm



- Sample a set of trajectories from $\pi_{\boldsymbol{\theta}}$
- Compute:

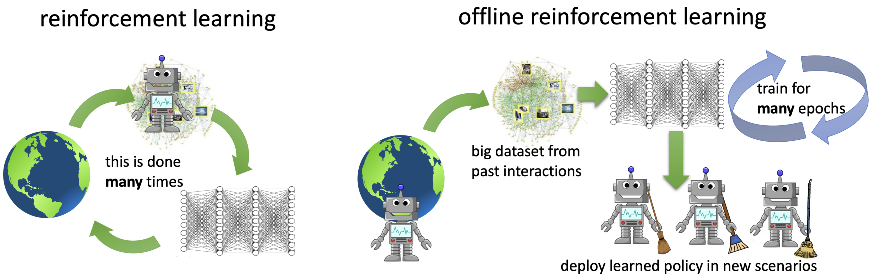
$$Loss(\boldsymbol{\theta}) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{t=1}^{H} \log \pi_{\boldsymbol{\theta}}(\mathbf{a}_t^{(i)}|\mathbf{s}_t^{(i)})R(\tau^{(i)}) \quad (2)$$

- Minimize the loss
- Iterate: sample again

PG = RWR !

- Equation (1) is the same as (2)!
- But wait, the basic PG algorithm is on-policy, and RWR uses expert data in the first step! What's happening?
- My guess: An on-policy algorithm will work under an off-policy regime if the behavioral samples are not worse than the current policy
- See this blogpost for a wider perspective:
  Data-driven Deep Reinforcement Learning
  https://bair.berkeley.edu/blog/2019/12/05/bear/

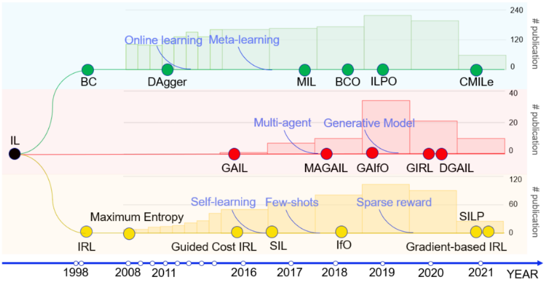
Offline RL



- If we perform just one iteration, this is offline reinforcement learning
- An open question is Under what condition on the batch data can we obtain an optimal policy this way?

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

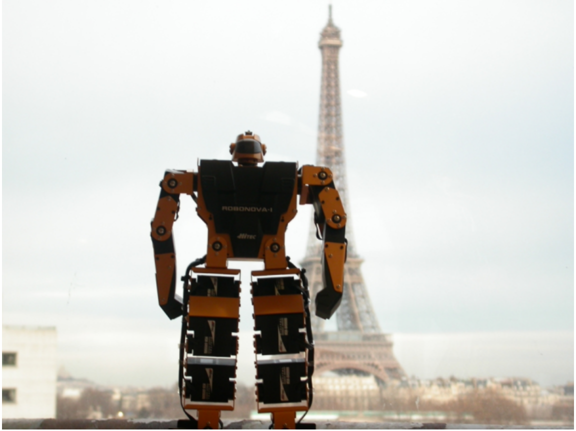
Imitation learning variants



- Learning from a single demonstration
- Learning without access to the actions
- Combined with MBRL, GCRL, HRL, ...

Zheng, B., Verma, S., Zhou, J., Tsang, I. W., and Chen, F. (2022) Imitation learning: Progress, taxonomies and challenges. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–16

Any question?



Send mail to: `Olivier.Sigaud@isir.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.

Zheng, B., Verma, S., Zhou, J., Tsang, I. W., and Chen, F. (2022).
Imitation learning: Progress, taxonomies and challenges.
IEEE Transactions on Neural Networks and Learning Systems, pages 1–16.