RL^2 (Fast RL via Slow RL)

https://arxiv.org/pdf/1611.02779

Paper Summary Notes

1 Main Ideas, High Level Assumptions

Context: recall the meta learning problem wants to improve sample efficiency on new tasks by learning a descriptive prior from different tasks, similar to how humans quickly learn new things by leveraging their general knowledge

- **idea**: cast the meta learning problem, (of developing a *fast* RL learning procedure) as an end-to-end RL problem. In other words, we view a good learning process of the agent *itself as an objective*, which can be optimized using standard reinforcement learning techniques
- **approach**:
  - encode the *fast* learning algorithm as the weights of an RNN which is trained by interacting with a distribution of MDP's.
  - the activations of the RNN store the state of the *fast* algorithm based on previous environment episodes, therefore functioning as a prior over the task
  - in this setup, the objective is to choose parameters of the RNN, such that the expected return from a sequence of interactions with an MDP is maximized

2 Approach and Formulation

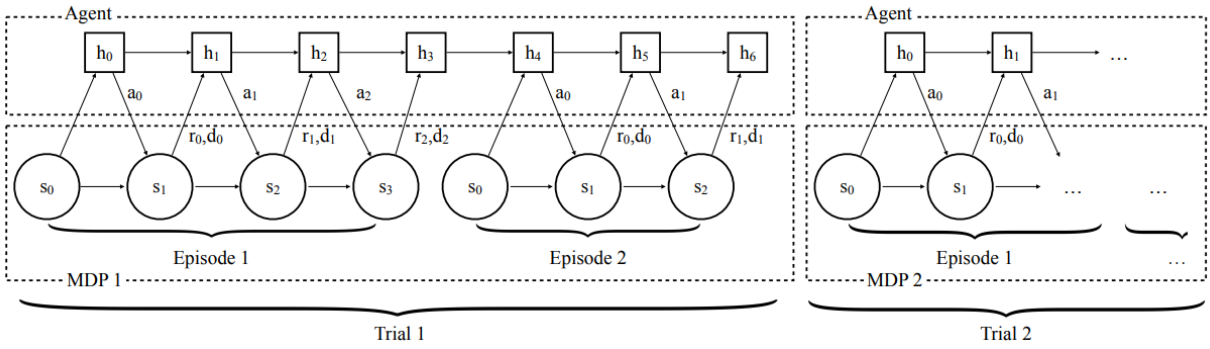


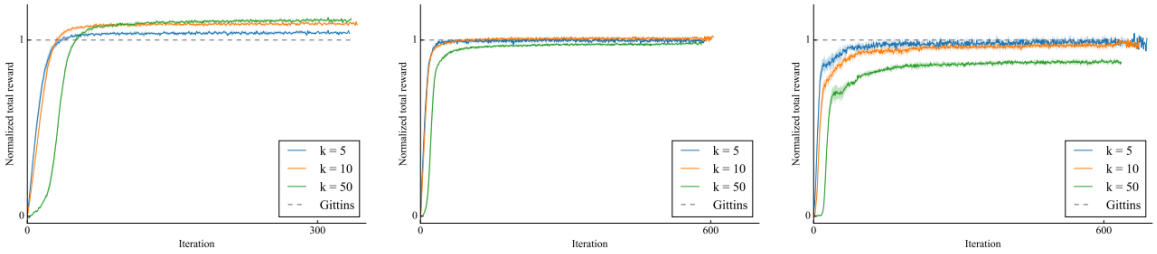
Figure 1: Procedure of agent-environment interaction

- hidden state of the policy is preserved to the next episode within a single MDP trail, but not preserved between trails
- intuitively, the hidden states capture the semantics of the learning procedure, and are therefore only reset after switching environments
- GRU cell is used for recurrent component to avoid vanishing gradients from long horizons
- output of GRU cell is fed through FC + softmax to produce output action distribution
- rewards are passed as *inputs* and trained across multiple MDP's

3 Results

3.1 Bandits

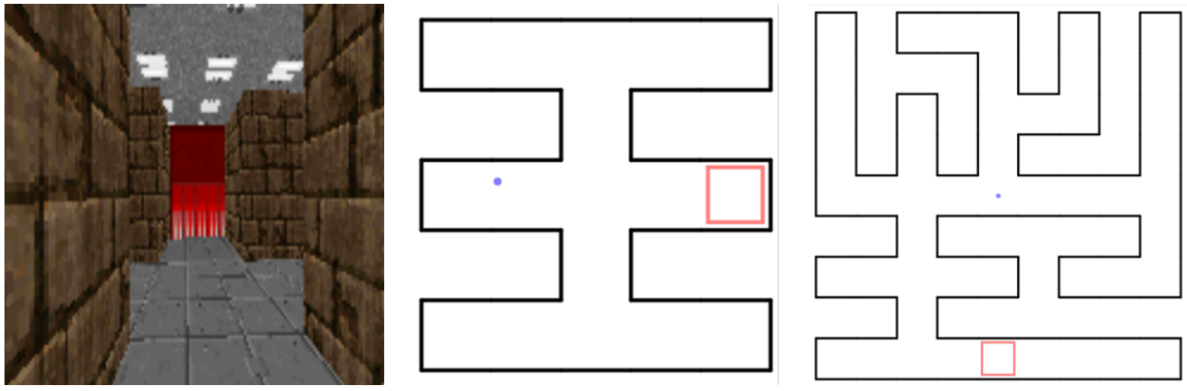
Setup	Random	Gittins	TS	OTS	UCB1	ϵ-Greedy	Greedy	RL^2
$n = 10, k = 5$	5.0	**6.6**	5.7	6.5	**6.7**	**6.6**	**6.6**	**6.7**
$n = 10, k = 10$	5.0	**6.6**	5.5	6.2	**6.7**	**6.6**	**6.6**	**6.7**
$n = 10, k = 50$	5.1	6.5	5.2	5.5	**6.6**	6.5	6.5	**6.8**
$n = 100, k = 5$	49.9	**78.3**	74.7	**77.9**	**78.0**	75.4	74.8	**78.7**
$n = 100, k = 10$	49.9	**82.8**	76.7	81.4	82.4	77.4	77.1	**83.5**
$n = 100, k = 50$	49.8	**85.2**	64.5	67.7	84.3	78.3	78.0	**84.9**
$n = 500, k = 5$	249.8	**405.8**	**402.0**	**406.7**	**405.8**	388.2	380.6	**401.6**
$n = 500, k = 10$	249.0	**437.8**	429.5	**438.9**	**437.1**	408.0	395.0	432.5
$n = 500, k = 50$	249.6	**463.7**	427.2	437.6	457.6	413.6	402.8	438.9



(a) $n = 10$ (b) $n = 100$ (c) $n = 500$

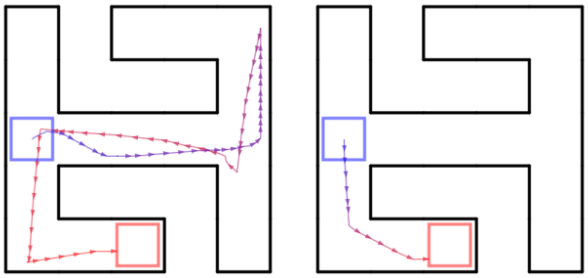
- k denotes number of bandits
- n denotes number of interactions
- scores normalized so that theoretically optimal bayesian method achieves 1, and random sampling achieves 0

3.2 Visual Navigation



(a) Sample observation (b) Layout of the 5×5 maze in (a) (c) Layout of a 9×9 maze

Figure 4: Visual navigation. The target block is shown in red, and occupies an entire grid in the maze layout.



(a) Good behavior, 1st episode (b) Good behavior, 2nd episode

- The *correct* learning technique would interact with a new maze such that on the first episode, we explore to find the objective, and on the second episode we take the shortest path

(a) Average length of successful trajectories

Episode	Small	Large
1	52.4 ± 1.3	180.1 ± 6.0
2	39.1 ± 0.9	151.8 ± 5.9
3	42.6 ± 1.0	169.3 ± 6.3
4	43.5 ± 1.1	162.3 ± 6.4
5	43.9 ± 1.1	169.3 ± 6.5

(b) %Success

Episode	Small	Large
1	99.3%	97.1%
2	99.6%	96.7%
3	99.7%	95.8%
4	99.4%	95.6%
5	99.6%	96.1%

(c) %Improved

Small	Large
91.7%	71.4%

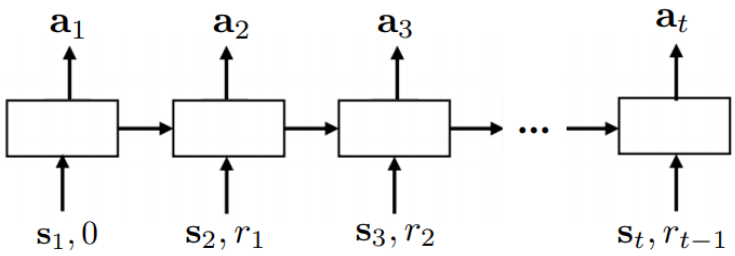
- there is significant reduction in trajectory lengths between the first two episodes, suggesting that the agent has learned how to use information effectively

4 Summary

- This method falls under the family of *black-box methods*

Black-box network
(LSTM, NTM, Conv, ...)

$$\mathbf{a}_t = f(\mathcal{D}_{\text{train}}, \mathbf{s}_t; \theta)$$



- Pros
  - general and expressive
  - variety of architecture choices for outer loop
- Cons:
  - hard optimization problem
  - no inductive bias of optimization built in (i.e. just let the RNN learn what it needs to learn)

5 Appendix

5.1 Formal Construction of Meta MDP

META-LEARNING AS RL

We now describe a way to construct an induced POMDP corresponding to a meta-learning problem. We assume knowledge of a set of MDPs, denoted by $\mathcal{M}$, and a distribution over them: $\rho_{\mathcal{M}} : \mathcal{M} \to \mathbb{R}_+$. This distribution should encode our prior knowledge about the structure of MDPs. We only need to sample from this distribution, and hence do not require tractable inference using $\rho_{\mathcal{M}}$. We also define n as the total number of episodes we will interact with a specific MDP.

To avoid confusion, we attach superscript to the specification of a single MDP, so that it becomes $M = (\mathcal{S}^M, \mathcal{A}^M, \mathcal{P}^M, r^M, \rho_0^M)$. For convenience, we drop γ and T from the specification and assume them to be consistent across all MDPs.

Before we give the formal definition, here is an intuitive description: at the beginning of each episode in this induced POMDP, we sample an MDP from $\rho_{\mathcal{M}}$, and let the agent interact with it for n episodes, where each episode has length up to T. The agent has access to the entire previous history, including actions, rewards, and terminal signals, and the history extends over multiple episodes in the MDP. The goal is to maximize the expected total discounted reward along this extended trajectory.

Formally, we construct a POMDP, $M(\mathcal{M}, \rho_{\mathcal{M}}, n)$, whose components are specified below:

- state space: $\mathcal{S}^{\mathcal{M}} = \mathcal{M} \times \mathcal{S} \times \mathcal{A} \times \mathbb{R} \times \{0, \dots, nT\}$;
- observation space: $\mathcal{O}^{\mathcal{M}} = \mathcal{S} \times \mathcal{A} \times \mathbb{R} \times \{0, 1\}$;
- action space: $\mathcal{A}^{\mathcal{M}} = \mathcal{A}$;
- transition function: $\mathcal{P}^{\mathcal{M}}((M', s', a'_{\text{prev}}, r'_{\text{prev}}, t') | (M, s, a_{\text{prev}}, r_{\text{prev}}, t), a) = \mathcal{P}^M(s'|s, a) \cdot \delta_a(a'_{\text{prev}}) \cdot \delta_{r(s,a)}(r'_{\text{prev}}) \cdot \delta_{t+1}(t') \cdot \delta_M(M')$;
- observation function: $o^{\mathcal{M}}((M, s, a_{\text{prev}}, r_{\text{prev}}, t)) = (s, a_{\text{prev}}, r_{\text{prev}}, 1\{t \equiv 0 \pmod T\})$;
- reward function: $r^{\mathcal{M}}((M, s, \cdot, \cdot, \cdot), a) = r^M(s, a)$;
- initial state distribution: $\rho_0^{\mathcal{M}}((M, s, a_{\text{prev}}, r_{\text{prev}}, t)) = \rho_{\mathcal{M}}(M) \cdot \rho_0^M(s) \cdot \delta_{a_{\text{empty}}}(a_{\text{prev}}) \cdot \delta_{r_{\text{empty}}}(r_{\text{prev}})$, where a_{empty} and r_{empty} are placeholders, typically chosen to be the zero vector (or scalar), and $\delta_x(x')$ is the Dirac delta function with the entire mass on x;
- discount factor: $\gamma^{\mathcal{M}} = \gamma$ assumed to be consistent across all MDPs;
- horizon: $T^{\mathcal{M}} = nT$.

The objective in this POMDP is again to maximize the expected total discounted reward. In the problems we consider, $\gamma^{\mathcal{M}}$ is treated a variance reduction parameter rather than the true objective, and it is assumed that the true metric we care about has discount 1. Maximizing this objective in the undiscounted case is equivalent to minimizing the cumulative pseudo-regret (Bubeck & Cesa-Bianchi, 2012), with an expectation taken over all possible MDPs. Different objectives can be considered in this meta-learning setup, such as risk-sensitive (Howard & Matheson, 1972) or minimax objectives (Strehl et al., 2009) more common in the frequentist literature. We leave these directions for future work.

The construction can be readily extended to the case where we have a distribution over POMDPs, rather than MDPs. In addition, the individual tasks may admit a termination function, which decides whether an episode can finish before the specified horizon T is reached. We leave these out of the formulation above to keep the notation from getting even more cluttered, and assume the MDP-based notation throughout.

REFERENCES

Sébastien Bubeck and Nicolo Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *arXiv preprint arXiv:1204.5721*, 2012.

Ronald A Howard and James E Matheson. Risk-sensitive markov decision processes. *Management science*, 18(7):356–369, 1972.

Alexander L Strehl, Lihong Li, and Michael L Littman. Reinforcement learning in finite mdps: Pac analysis. *Journal of Machine Learning Research*, 10(Nov):2413–2444, 2009.