VARIATIONAL INTRINSIC CONTROL

Karol Gregor, Danilo Jimenez Rezende and Daan Wierstra
DeepMind
{karolg,danilor,wierstra}@google.com

ABSTRACT

We introduce a new unsupervised reinforcement learning method for discovering the set of *intrinsic* options available to an agent. This set is learned by maximizing the number of different states an agent can reliably reach, as measured by the mutual information between the set of options and option termination states. To this end, we instantiate two policy gradient based algorithms, one that creates an explicit embedding space of options and one that represents options implicitly. Both algorithms also yield a tractable and explicit empowerment measure, which is useful for empowerment maximizing agents. Furthermore, they scale well with function approximation and we demonstrate their applicability on a range of tasks.

1 INTRODUCTION

What should an agent do in the absence of external rewards? A common view is that agents should seek out novelty and try to learn a model of the environment. This is formally understood as maximizing information gain, as measured by the difference in compression of its experience before and after learning (Schmidhuber, 1991; 2010; Bellemare et al., 2016; Houthooft et al., 2016). However, the world can be complex and hard to model, and such a strategy might not translate into the ability to control the environment and reach goals. An alternative strategy is to focus on aspects of the environment that can be controlled by the agent. This notion of controllability can be measured as the number of states that can be reliably reached by the agent within some time horizon. In this work we provide algorithms for learning a set of closed loop options that maximize controllability.

This work builds on the empowerment literature (Salge et al., 2014; Klyubin et al., 2005; Blahut, 1972; Arimoto, 1972; Mohamed & Rezende, 2015) as it optimizes the same measure – the mutual information between action choices and future states. However, it differs in important ways. First, we utilize full state-conditional policies and show that these are critical for good control and reaching higher levels of empowerment. Second, unlike most previous approaches (except, to an extent (Mohamed & Rezende, 2015)), our algorithm scales to noisy, visually complex domains by leveraging deep learning techniques for function approximation.

Another related approach to unsupervised control was proposed in (Oudeyer & Kaplan, 2009; Baranes & Oudeyer, 2013). One important difference between their work and ours is that we don't require that a low dimensional option embedding space is given, rather we learn it from raw observations by maximizing controllability.

We begin presenting the algorithm by defining an option as an element Ω of a space and an associated policy $\pi(a|s,\Omega)$ that chooses an action a in a state s when following Ω. The policy π has a special *termination* action that terminates the option and yields a final state s_f. For example, 1) Ω takes a finite number of values $\Omega \in \{1,\dots,n\}$. This is the simplest case in which for each i a separate policy π_i is followed. 2) Ω is a binary vector of length n. This captures a combinatorial number 2^n of possibilities. 3) $\Omega \in R^d$ is a d-dimensional real vector. Here the space of options is smooth, so nearby policies behave similarly.

To maximize controllability we would like different options Ω to reach different final states s_f and be able to reach a large number of such states. Thus we want to maximize the entropy of the distribution $p(s_f|s_0)$ of states reached, but for a given Ω reach a small number of states, minimizing the entropy of $p(s_f|s_0,\Omega)$. This is precisely the mutual information between options and final states, which is the quantity we want to maximize. In this work we will instead maximize a more tractable variational lower bound on this:

$$I^{VB}(\Omega, s_f|s_0) = -\sum_{\Omega} p^C(\Omega|s_0)\log p^C(\Omega|s_0) + \sum_{\Omega,s_f} p^J(s_f|s_0,\Omega)p^C(\Omega|s_0)\log q(\Omega|s_0,s_f). \quad (1)$$

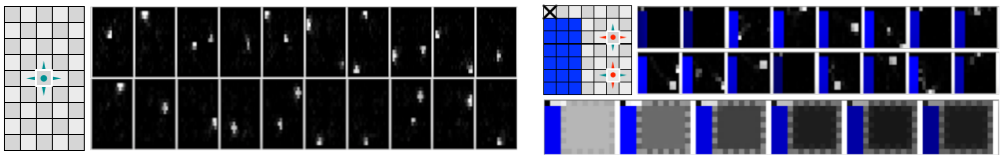
Here $p^C(\Omega|s_0)$ denotes the distribution of options that yields maximum coverage of the state space. The second term ensures that different options achieve different states, whereas the first term ensures that a wide range of Ωs is used. For a given option Ω, the distribution of states s_f reached is expressed by $p^J(s_f|s_0,\Omega)$. q tries to infer which option was followed. It can infer this only if the agent is able, for different Ω's, to reach different final states s_f.

These considerations suggest the following algorithm for maximizing this objective: 1) pick option Ω from $p^C(\Omega|s_0)$ and follow the policy $\pi(a|s,\Omega)$ reaching a state s_f; 2) measure how well this Ω can be inferred from s_f relative to Ω prior; this is our intrinsic reward $r_I = \log q - \log p^C$; 3) reinforce the actions chosen by the policy; 4) reinforce the option chosen by p^C. At the same time learn the option embedding function q by regressing it towards the option followed. This procedure is summarized in Algorithm 1 and in fact can be directly derived by differentiating the objective (Eq. 1).

Algorithm 1 Intrinsic Control with Explicit Options

Assume an agent in a state s_0
for episode = $1, M$ **do**
  Sample $\Omega \sim p^C(\Omega|s_0)$
  Follow policy $\pi(a|\Omega, s)$ till termination state s_f
  Regress $q(\Omega|s_0, s_f)$ towards Ω
  Calculate intrinsic reward $r_I = \log q(\Omega|s_0, s_f) - \log p^C(\Omega|s_0)$
  Use a reinforcement learning algorithm update for $\pi(a|\Omega, s)$ to maximize r_I.
  Reinforce option prior $p^C(\Omega|s_0)$ based on r_I.
  Set $s_0 = s_f$
end for
Note: Empowerment at s is estimated by the reinforce baseline of p^C, which tracks r_I.

We train an agent using this algorithm on a gridworld with a discrete space of options (figure below). Different options learned to terminate at different localized points in the environment. This is shown in the figure below on the left, where each rectangle corresponds to a different option and the bright spots indicate the locations where q is large. These are the locations the agent attempts to get to. We also trained the algorithm on a modified 'dangerous' version of the gridworld, where the agent has to take an action from a correct subset at different points in the environment, otherwise it would fall into a state where it is stuck for a long time. Additionally the actions are noisy. With the open loop algorithms considered in previous works, the agent is unable to navigate in this environment. The bottom right of the Figure shows the exact open loop empowerment at different locations of the environment for increasing option lengths, demonstrating that the empowerment decreases with option length for these policies. In contrast, our agent learns to safely navigate this environment.



While algorithm 1 is formulated for an arbitrary option space, optimization proved hard in practice when using a continuous option space. This might be remedied with methods such as experience replay which are routinely used to stabilize reinforcement learning algorithms (Mnih et al., 2015).

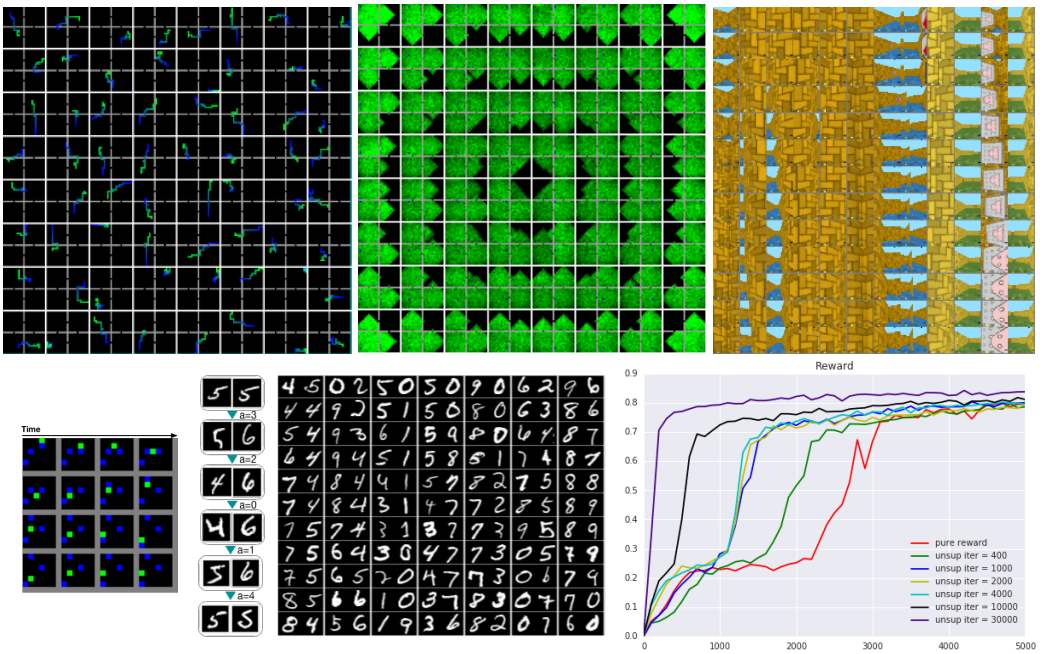
We now turn to the second algorithm, which trains with much less difficulty and is able to operate on a complex three dimensional simulated environment using raw pixels. The basic idea is to stay in the action space but consider closed loop policies. The option Ω of the previous section defines

Algorithm 2 Intrinsic Control with Implicit Options

Full update
Follow policy $\pi^p(a_t|s_t^p)$, $s_t^p = f^p(s_{t-1}^p, x_t, a_{t-1})$ resulting in experience $x_0, a_0, \ldots, x_f$.
For each t, regress policy $\pi^q(a_t|s_t^q)$, $s_t^q = f^q(s_{t-1}^q, x_t, a_{t-1}, x_f)$ towards action a_t
Calculate intrinsic reward $r_I = \sum_t \log \pi^q(a_t|s_t^q) - \log \pi^p(a_t|s_t^p)$
Reinforce the policy π^p with r_I.
Exploratory update
Follow policy $\pi^p(a_t|s_t^p)$, $s_t^p = f^p(s_{t-1}^p, x_t, a_{t-1})$ with exploration (ϵ or other) resulting in experience $x'_0, a'_0, \ldots, x'_f$.
For each t, regress policy $\pi^q(a'_t|s_t^q)$, $s_t^q = f^q(s_{t-1}^q, x'_t, a'_{t-1}, x'_f)$ towards action a'_t
Note: Empowerment is estimated by the reinforce baseline of π^p.

our action choices at different time steps, which depend on what happened in the environment at the previous steps. The resulting algorithm is shown in Algorithm 2.

All policies are parameterized by MLPs followed by an LSTM. We demonstrate that the algorithm learns a good control policy on a number of environments. The results are shown in the figures below. Reading along the rows from top left to bottom right: 1) Learned trajectories in a 25×25 four room gridworld. They are extended unlike trajectories produced by a random motion. 2) Distribution of end states from different starting positions for options of length 25. The learned policies result in a distribution of end states that is uniform among reachable states. 3) Visually complex three dimensional environment where the agent achieves a controllability score of $\exp(5.4) = 221$ states. 4) Example trajectory in a 6×6 world containing three movable blocks that can be pushed around by the agent. During a given trajectory, the agent pushes different blocks to different locations and on average achieves a controllability of approximately 1200 states. 5) This environment consists of pairs of MNIST digits which the agent can control with 5 actions (increment, decrement, maintain class label). Transitions are stochastic as the digits are resampled at every step from the current class. The agent only sees an image containing both digits at every step. Thus the inputs are noisy. The control reached is nearly all of the 100 possible control states, showing that the agent can focus on the controllable part of the space and ignore the uncontrollable noise. 6) An agent is allowed to learn to control a 15×15 four room gridworld and after a period of time, a reward is turned on. The agent is able to find the reward faster with a longer pre-training period.



REFERENCES

Suguru Arimoto. An algorithm for computing the capacity of arbitrary discrete memoryless channels. *IEEE Transactions on Information Theory*, 18(1):14–20, 1972.

Adrien Baranes and Pierre-Yves Oudeyer. Active learning of inverse models with intrinsically motivated goal exploration in robots. *Robotics and Autonomous Systems*, 61(1):49–73, 2013.

Marc G Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. *arXiv preprint arXiv:1606.01868*, 2016.

Richard Blahut. Computation of channel capacity and rate-distortion functions. *IEEE transactions on Information Theory*, 18(4):460–473, 1972.

Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. Variational information maximizing exploration. *arXiv preprint arXiv:1605.09674*, 2016.

Alexander S Klyubin, Daniel Polani, and Chrystopher L Nehaniv. Empowerment: A universal agent-centric measure of control. In *2005 IEEE Congress on Evolutionary Computation*, volume 1, pp. 128–135. IEEE, 2005.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

Shakir Mohamed and Danilo Jimenez Rezende. Variational information maximisation for intrinsically motivated reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 2125–2133, 2015.

Pierre-Yves Oudeyer and Frederic Kaplan. What is intrinsic motivation? a typology of computational approaches. *Frontiers in neurorobotics*, 1:6, 2009.

Christoph Salge, Cornelius Glackin, and Daniel Polani. Empowerment–an introduction. In *Guided Self-Organization: Inception*, pp. 67–114. Springer, 2014.

Jürgen Schmidhuber. Curious model-building control systems. In *Neural Networks, 1991. 1991 IEEE International Joint Conference on*, pp. 1458–1463. IEEE, 1991.

Jürgen Schmidhuber. Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE Transactions on Autonomous Mental Development*, 2(3):230–247, 2010.