InfoMARL: cooperation and competition via signalling and hiding intentions

DJ Strouse

January 26, 2018

1 Introduction

We propose to facilitate cooperation / competition between agents in a multi-agent RL (MARL) setting by encouraging agents to signal / hide their intentions. In the setting we consider, the first agent has access to the goal information and seeks to maximize its own reward, along with an information-theoretic regularizer meant to encourage its behavior to either reveal or hide the private information it has about the goal. A second agent is then trained to maximize its own reward only from observations of the first agent. This second agent does not have access to the goal information and must thus infer the goal from the actions of the first agent who does. The advantage of our approach is that the first agent can be trained without a model of nor interaction with the second agent.

2 Regularizing policies with information

In a typical (multi-goal) RL setup, an agent seeks to maximize its expected reward $\mathbb{E}[r]$, where the expectation is taken over episodes, goals, time, the dynamics of the environment, and the agent's policy. It is common to add auxiliary losses, such as the entropy of the policy to encourage exploration [Mnih et al., 2016], or pixel prediction/control to encourage exploration in the face of sparse rewards [Jaderberg et al., 2016]. Here we propose to add the mutual information between goal and action (conditioned on state), $I(A; G \mid S)$, where G is the goal for the episode, A is the chosen action, and S is the state of the agent. That is, we will train agents to optimize the loss:

$$L = -\mathbb{E}[r] + \beta I(A; G \mid S)\,, \tag{1}$$

where β is a tradeoff parameters whose sign determines whether we want our agents to signal (negative) or hide (positive) their intentions, and whose magnitude determines the relative preference for rewards and intention signalling/hiding.

The information $I(A; G \mid S)$ is a functional of the multi-goal policy $\pi_g(a \mid s) = p(a \mid s, g)$, that is the probability distribution over actions given the current goal and state, and is given by:

$$I(A; G \mid S) = \sum_s p(s)\, I(A; G \mid S = s) \tag{2}$$

$$= \sum_s p(s) \sum_{a,g} p(a, g \mid s) \log \frac{p(a, g \mid s)}{p(a \mid s)\, p(g \mid s)}. \tag{3}$$

It will be useful to rewrite the above into a form in which the dependence on the policy is explicit and the sums are over episodes, so that we can estimate and optimize from samples. Rewriting $p(a, g \mid s)$ inside the log as $p(a, g \mid s) = p(a \mid s, g)\, p(g \mid s) = \pi_g(a \mid s)\, p(g \mid s)$ and cancelling the $p(g \mid s)$ in the numerator and denominator, we have:

$$I(A; G \mid S) = \sum_s p(s) \sum_{a,g} p(a, g \mid s) \log \frac{\pi_g(a \mid s)}{p(a \mid s)}. \tag{4}$$

Noting that $p(a, g \mid s)\, p(s) = p(g)\, p(s \mid g)\, p(a \mid s, g) = p(g)\, p(s \mid g)\, \pi_g(a \mid s)$, we can rearrange the terms outside the log to arrive at:

$$I(A; G \mid S) = \sum_g p(g) \sum_s p(s \mid g) \sum_a \pi_g(a \mid s) \log \frac{\pi_g(a \mid s)}{p(a \mid s)}. \tag{5}$$

At this point, we can see that the quantity involving the sum over actions is a KL divergence between two distributions: the goal-dependent policy $\pi_g(a \mid s)$ and a goal-independent policy $p(a \mid s)$. This goal-independent policy comes from marginalizing out the goal, that is $p(a \mid s) = \sum_g p(g)\, \pi_g(a \mid s)$. Whye Teh et al. [2017] independently introduced this regularizer (without noting the connection to the mutual information) for an alternative reason - they wanted to encourage an agent to transfer knowledge between tasks (goals). To maintain consistently with their notation, we will denote $\pi_0(a \mid s) \equiv p(a \mid s)$ and refer to it as the "base policy," whereas we will refer to $\pi_g(a \mid s)$ as simply the "policy." Thus, we can rewrite the information above as:

$$I(A; G \mid S) = \sum_g p(g) \sum_s p(s \mid g)\, \mathrm{KL}[\pi_g(a \mid s) \mid \pi_0(a \mid s)]\,. \tag{6}$$

Writing the information this way suggests a method for stochastically optimizing it. First, we sample a goal g from $p(g)$, that is we initialize an episode of some task. Next, we sample states s from $p(s \mid g)$, that is we generate state trajectories using our policy $\pi_g(a \mid s)$. For each step of the trajectory, we take the gradient of the KL with respect our policy $\pi_g(a \mid s)$ and adjust the policy in that direction.

While there are many approaches to maximizing expected reward, because the information is a functional of the policy and we wish to simultaneously optimize rewards and information, it will be most natural to take a policy gradient approach. That is, the gradient of our loss (with respect to policy parameters θ) at time step t in an episode with goal g will given by:

$$\nabla_\theta L_t = (\nabla_\theta \log \pi_g(a_t \mid s_t))\, R_t + \beta \nabla_\theta \mathrm{KL}[\pi_g(a_t \mid s_t) \mid \pi_0(a_t \mid s_t)]\,, \tag{7}$$

where $R_t = \sum_{\tau=t}^{T} \gamma^{\tau-t} r_t$ is the discounted return, T is the length of the episode, γ the discount factor, and r_t the reward at time step t. By using the actual episode return R_t, we are assuming finite episodes and taking an approach known as the REINFORCE algorithm [Williams, 1992]. However, one could replace R_t with an estimated return such as a learned Q-value $Q(s_t, a_t)$.

In summary, the implementation of our approach simply involves adding $\beta \nabla_\theta \mathrm{KL}$ to the policy gradient in a standard implementation of REINFORCE.

References

M. Jaderberg, V. Mnih, W. M. Czarnecki, T. Schaul, J. Z Leibo, D. Silver, and K. Kavukcuoglu. Reinforcement Learning with Unsupervised Auxiliary Tasks. *ArXiv e-prints*, November 2016.

V. Mnih, A. Puigdomènech Badia, M. Mirza, A. Graves, T. P. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu. Asynchronous Methods for Deep Reinforcement Learning. *ArXiv e-prints*, February 2016.

Y. Whye Teh, V. Bapst, W. M. Czarnecki, J. Quan, J. Kirkpatrick, R. Hadsell, N. Heess, and R. Pascanu. Distral: Robust Multitask Reinforcement Learning. *ArXiv e-prints*, July 2017.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3-4):229–256, May 1992. ISSN 0885-6125. doi: 10.1007/BF00992696. URL https://doi.org/10.1007/BF00992696.