Skill Learning for Robots via Temporal Variational Inference

Xiaoyang Shi

October 2021

I am fascinated by the question of how to enable robots to learn. Human brains are complicated machines. Watching a child growing up at home, I am constantly amazed by how she is able to rapidly acquire an increasing number of skills, mastering an exponential amount of sophisticated tasks. However, the truth regarding why she learns so fast is that when she encounters a new task, the skills learned from her previous experience get reused. In comparison, nowadays robots mostly still have to learn everything from scratch - they have no "skills". Thus, I wonder, is there a way for machine learning agents to emulate how humans perform tasks by composing a variety of skills? For example, a robot equipped with reaching, grasping, and pouring skills could compose them to make a cup of tea as well as pour cereal into a bowl. In other words, is there a powerful algorithm that enables robots to precisely do that?

This is not an easy task. There is no clear-cut way of even defining a "skill", other than that it elicits a certain pattern of actions in response to a particular situation. The traditional source of learning data would be human demonstrations, but it becomes too laborious to segment those demonstration trajectories into semantically meaningful skills by hand. Last but not least, though the ability to compose skills leads to a combinatorial increase in the robot's capabilities, how could a robot know the "right" sequence of using such skills in the face of a new task?

Hence, I was extremely excited when I saw the paper "Learning Robot Skills with Temporal Variational Inference" by Shankar et al. published on the International Conference on Machine Learning (ICML 2020). The paper proposes an algorithm that jointly learns skills and how to compose and use them from demonstrations in an unsupervised manner. At the core of the algorithm is the method of temporal variational inference (TVI) based on the factorization of trajectory likelihoods, learning low-level control policies and high-level policies for skill selection at the same time.

Particularly, the algorithm adopts a latent variable representation of skills; consequently, the problem of inferring skills can be treated as inferring latent variables, which could then be accomplished via temporal variational inference. Optimizing the objective of selecting and sequencing skills to reconstruct the original trajectory with respect to the distributions leads to low and high level policies of skills.

As preliminaries, the method adopts an undiscounted Markov Decision Process without rewards (MDP \R), which is a tuple that consists of states $s \in S$, actions $a \in A$ and a transition function between successive states $P(s_{t+1}|s_t, s_t)$. The goal of learning is to acquire a policy $\pi(s|a)$ that informs the robot what action to take after observing any state.

They use an Evidence Lower Bound (ELBO) objective to maximize the log-likelihood of trajectories

$\mathcal{L} = \mathbb{E}_{\tau \sim \mathcal{D}}[\log p(\tau)]$ across the dataset under the learned policy. I summarize the core algorithm below:

Algorithm 1 Temporal Variational Inference for Learning Skills (TVI)

Key idea: Infer skills as latent variables based on the factorization of trajectory likelihoods; jointly learn low-level control policies and high-level policies for skill selection.

Input: Demonstration dataset $\mathcal{D}$

Output: Low-level policy π, high-level policy η

Initialize π_θ, η_ϕ, q_ω as LSTMs
Pretrain π with VAE
for $iter \in [1, 2, \ldots, N_{iterations}]$ **do**
    for $i \in [1, 2, \ldots, |\mathcal{D}|]$ **do**
        $\tau_i \leftarrow \mathcal{D}$ ▷ Retrieve trajectory from dataset
        $\zeta \sim q(\zeta|\tau_i)$ ▷ Sample latent sequence from variational network
        $J \leftarrow \sum_t \log \pi(a_t|s_{1:t}, a_{1:t-1}, \zeta_{1:t}) + \sum_t \log \eta(\zeta_t|s_{1:t}, a_{1:t-1}, \zeta_{1:t-1}) - \log q(\zeta|\tau)$ ▷ Evaluate likelihood objective under current policy estimates
        Update π_θ, η_ϕ, q_ω via $\nabla_{\theta,\phi,\omega} J$
    end for
end for

In my opinion, what's interesting about their method is that 1) in their hierarchical setup, the agent has two policies, such that the high-level controller (like the brain) abstracts away the details of low-level control and reasons over high-level skills instead, and consequently a robot can address more complex and longer term tasks; 2) Learning skills from demonstrations in an unsupervised fashion is appealing, since it allows non-robotics-expert humans to demonstrate solving the target tasks, bypassing the tedium of manually specifying these skills or carefully engineering solutions to the tasks; and 3) Using temporal variational inference allows the agent to construct an objective that directly affords it usable policies on optimization (there is no additional post hoc policy learning step).

After reading the paper, I thought of several potential directions for future work. For humans, we learn skills, but we also set up a hierarchy of skills - for instance, playing badminton is a skill that requires sub-skills such as running and jumping, while playing tennis is another skill that requires running and jumping; however, there is also something unique to either playing badminton or playing tennis. By composing learned sub-skills into high-level skills, robots can potentially learn to accomplish a more diverse set of tasks. Thus, I think to learn more effectively, a better algorithm should be able to capture the hierarchical nature of skills. Also, continuing on the last example, one cannot possibly learn to play badminton without knowing how to run or how to jump; hence, there seems to be a temporal sequence of learning skills in nature (or some may say learn easy things first), which is not present in the algorithm proposed in this paper. A more advanced algorithm could opt to incorporate curriculum learning into skill learning. Another direction is visual skill learning: since learning from demonstration data in this algorithm still requires specific state/action pairs, learning directly from visual observations (pixel data) would be more challenging yet corresponds better to the real-world scenario. This might be addressed by at least integrating more "powerful" recurrent models such as the Transformer, which is now revolutionizing the world of natural language processing.