Task-Informed Embedded State Transitions for Model-Based Reinforcement Learning

Ryan Theisen

***Abstract*— We explore a technique to model state transitions for use in model-based reinforcement learning. We use a latent encoding of the state space to 1) model a transition distribution between states conditioned on actions and 2) predict a distribution over rewards given a transition realization. We show that, in addition to being used in a fully model based setting, this method can be used to augment a variety of different reinforcement learning techniques.**

I. INTRODUCTION

In model based reinforcement learning (RL), the representative agent is assumed to have full knowledge of the state-transition distribution $p(s'|s,a)$. For example, in Q-learning, the agent can learn the optimal policy π by iteratively applying the Bellman backup equation:

$$Q(s,a) = R(s) + \gamma \mathbb{E}_{s' \sim p(s'|s,a)}[\text{argmax}_{a'} Q(s',a')]$$

where $\pi(s) = \text{argmax}_a Q(s,a)$. However, in most real-world cases, the state transition dynamics are unknown, leading to alternate approaches to the RL problem, which are unable to make use of the information provided by $p(s'|s,a)$.

We propose a new method for estimating the state transition distribution, except defining the transition instead on a latent representation $z \in \mathcal{Z}$ of the actual state $s \in \mathcal{S}$. Approaches to encoding input in such a way have been widely studied in the machine learning literature, and have been shown to be useful estimators of the latent data manifold. For examples, in variational autoencoders (VAEs), the latent distribution $p(z|x)$ is derived by minimizing a variation lower bound of the data likelihood. We employ a similar approach to encode the state s into a latent distribution $p(z|s)$, which is in turn used to estimate a distribution $p(z'|z,a)$ that approximates the transition dynamics $p(s'|s,a)$. Furthermore, we model the reward received after transitioning to a new state as a distribution $p(r|z')$.

II. BACKGROUND

to do

III. TASK-INFORMED EMBEDDED STATE TRANSITIONS

A. General Method

Consider the common task of modeling an observed variable X in terms of a latent representation Z. There are many ways to characterize what properties such a representation should have, though a reasonable approach, and the one we employ in this paper, is defined by the notions of *sufficiency* and *minimality*. Sufficiency, in particular, is relative to a particular task; that is, for a given task Y, the representation Z should encode all the necessary information about X to accomplish Y. We can express this formally in terms of mutual information via the condition: $I(X;Y) = I(Z;Y)$. Minimality, in turn, means that Z should contain as little information from X as possible while be sufficient for the given task. Formally, we would like that $I(X;Z)$ is a small as possible while having Z be sufficient. Noting that $I(X;Y) - I(Z;Y) = H(Y|Z) - H(Y|X)$, we can write this in terms of a constrained optimization problem:

$$\begin{aligned} &\text{minimize } I(X;Z) \\ &\qquad \text{s.t. } H(Y|X) = H(Y|Z) \end{aligned}$$

Such an objective has been studied previously in the literature (Achille, Soatto 2017; Tibshy 2015), and is known as the Information Bottleneck Principal, with the associated objective:

$$\mathcal{L} = H(Y|Z) + \beta I(X;Z)$$

In this paper, we study a particular case of this problem, where the task is twofold: (i) recovering the reward r and (ii) modeling the transition dynamics $p(s'|s,a)$.

Ignoring task (ii) for the moment, we see task (i) is a straightforward example of the above objective, in which we have:

$$\mathcal{L} = H(r|z) + \beta I(s;z)$$

To consider task (ii), we first consider a sample trajectory (s,a,s',r), in particular noting that $s' \sim p(s'|s,a)$. We would like to have an encoding $p(z'|z,a)$ that models this distribution as closely as possible. In particular, we consider the following addition to the above objective:

$$D_{KL}(p(z'|s') \| p(z'|z,a))$$

That is, the latent transition distribution for the next state given the current state and $a \in \mathcal{A}$ should "look like" the latent distribution for the next state (which is by nature conditioned on the previous state and action). Then we define the aggregate objective as follows:

$$\mathcal{L} = H(r|z') + \beta I(s;z) + \lambda D_{KL}(p(z'|s') \| p(z'|z,a))$$

Note that the encodings $p(z'|s')$ and $p(z|s)$ share the same generating function (a neural network), simply conditioned on different states. See Figure 1 for a diagram of the final model.

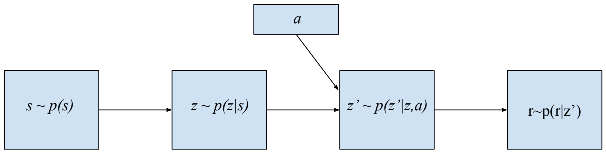

Fig. 1. A diagram of the latent transition model. In practice, the encoder $p(z|s)$, transition $p(z'|z, a)$, and reward $p(r|z')$ are parameterized as neural networks that output the parameters of some distribution (e.g. a Gaussian).

IV. Example: Model-Based Q-Learning

A. Monte Carlo Model-Based Q Learning

Consider the standard Bellman Equation:

$$Q(s,a) = R(s) + \gamma \mathbb{E}_{s' \sim p(s'|s,a)}[\text{argmax}_{a'} Q(s',a')]$$

Under the hypothesis that the representation z is sufficient for the given task, we can reframe this problem in terms of the latent encoding z:

$$\begin{aligned} Q(z,a) &= R(z) + \gamma \mathbb{E}_{z' \sim p(z'|z,a)}[\text{argmax}_{a'} Q(z',a')] \\ &= R(z) + \gamma \int_{\mathcal{Z}} \text{argmax}_{a'} Q(z',a') p(z'|z,a) dz' \end{aligned}$$

Since $p(z'|z, a)$ is known, we can estimate this integral using a Monte Carlo approach:

$$\begin{aligned} Q(z,a) &= R(z) + \gamma \int_{\mathcal{Z}} \text{argmax}_{a'} Q(z',a') p(z'|z,a) dz' \\ &\approx R(z) + \frac{\gamma}{N} \sum_{k=1}^{N} \text{argmax}_{a'} Q(z'_k, a') \end{aligned}$$

where $z'_1, ..., z'_N \sim p(z'|z, a)$ and $R(z) \sim p(r|z)$.

Using this set up, we can estimate Q using a modified version of the DQN algorithm, minimizing the objective

$$\mathcal{L} = \frac{1}{2}\Big(Q(z,a) - \big(R(z) + \frac{\gamma}{N} \sum_{k=1}^{N} \text{argmax}_{a'} Q(z'_k, a')\big)\Big)^2$$

Note that, unlike the standard DQN algorithm, given the distribution $p(z'|z, a)$ and the function $R(z)$, this formulation can be trained entirely off-line, with no active interaction with the environment.

V. Experiments

VI. Conclusion and Future Work