CS594 Project Proposal

JACK YANSONG LI Email: yli340@uic.edu
SHUO WU Email: swu99@uic.edu

1 Problem Formulation

We consider an agent with state $s \in \mathcal{S}$ and action $a \in \mathcal{A}$. The agent interact with a changing environment. Our goal is to train a control policy π that is adaptive to the changing environment.

However, it is hard to model the entire changing environment and its dynamics. So, we choose some features and pack them into a large vector $e_t \in \mathcal{E}$. We call e_t the *explicit environment feature* at time t. Note that our policy π_t should also depends on e_t, for a large e_t, this will require a very large computational resources. Moreover, in the real application, the feature e_t is usually unkown to us. To solve the issue, we consider the algorithm given in [1]. The algorithm can be summarized as the following procedure.

- Phase 1: Given e_t, use policy class as

$$a_t = \pi(s_t, \mu(e_t); \theta_\pi),$$

  optimize over μ, θ_π with respect to a state value function $V^{\pi(\theta_\pi, \theta_\phi)}(s_0)$.

- Phase 2: Replace $\mu(e)$ with $\phi(s_{t-H:t}, a_{t-H:t-1}; \theta_\phi)$. Find θ_ϕ to minimize

$$\mathbb{E}_e \|\mu(e) - \phi(\ldots; \theta_\phi)\|_2^2.$$

Briefly speaking, in phase 1, we encode e_t with an encoder μ to reduce its dimension. We call $z_t = \mu(e)$ a *latent environment feature*. We optimize μ and θ_π with respect to a given value function. In phase 2, we applied linear regression with the history gained by phase 1 to train an encoder ϕ that is close to μ. However, this ϕ takes history as input to give an latent environment feature $\hat{z}_t = \phi(s_{t-H:t}, a_{t-H:t-1}; \theta_\phi)$ that is close to z_t. We can futhter abstract thier goal into the following:

- Find the best θ_π, θ_ϕ for $\pi(s_t, \phi(s_{t-H:t}, a_{t-H:t-1}; \theta_\phi); \theta_\pi)$

$$\underset{\theta_\pi, \theta_\phi}{\text{maximize}} \quad J(\theta_\pi, \theta_\phi) \triangleq \mathbb{E}_e[V^{\pi(\theta_\pi, \theta_\phi)}(s_0)].$$

However, why introduce an extra encoder μ instead of optimize θ_π and θ_ϕ from history directly? Will an extra encoder boosts or stabilizes the training procedure? This is the question we are interested.

2 Toy Case Experiment

The first thing we need to do is to build a toy case which we can test these two different setting.

2.1 Two encoder

2.1.1 Phase 1

$$x_{t+1} = A(e)x_t + B(e)u_t, \quad e \in \mathcal{E},$$

where

$$\mathcal{E} = \{e_1, e_2, \ldots, e_N\}.$$

We define $A(e)$ as

$$A(e) = \begin{pmatrix} e_1^2 + \cdots + = a_{11}(e) & \ldots \\ \cdots & \cdots \end{pmatrix}.$$

The cost function is defined as

$$\sum (x_t^\top Q x_t + u_t^\top R u_t).$$

The μ and π is linear parameterized.

2.1.2 Phase 2

Given $(x_1, x_2, \ldots, x_H)$

$$J(A, B) \triangleq \sum \|x_t - \hat{x}_t\|_2^2$$

where

$$\hat{x}_{t+1} = A\hat{x}_t + Bu_t$$

$$(A^\star, B^\star) = \operatorname*{argmin}_{A,B} J(A, B)$$

2.2 Only 1 encoder

Same problem formulation with ϕ and π be linear parameterized.

Bibliography

[1] Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. RMA: Rapid Motor Adaptation for Legged Robots. jul 2021.