1 Introduction and preliminaries

1. The key idea is to use a real-life trial to evaluate a policy, but then use the simulator (or model) to estimate the derivative of the evaluation with respect to the policy parameters.

2. Argues that this approach does not require a "correct" model of the MDP. What I think they mean is that the model does not have to be correct enough so that it can be used for planning.

2 Key ideas

1. We will obtain the policy by the policy gradient. In REINFORCE, the policy gradient is obtained from the log likelihood trick.

2. The log likelihood trick is useful because we do not have to know the transition dynamics, which the expectation of trajectory is over. This gradient estimate is unbiased but has high variance.

3. If we have a learnt transition model, then we can differentiate through the model to obtain a **biased** gradient estimate.

4. The error of this gradient estimate comes from two sources: the functional form of the gradient and where it is evaluated at.

5. We can remove the second source of error by evaluating the gradient obtained from differentiating the model at a real trajectory, generated by running the current policy in the real MDP.

6. So long as the estimated gradient is within 90 degree of the true gradient, taking a small step in the direction of the estimated gradient will improve the policy.

3 Formal description of key idea

1. Let $f_t(s, a)$ gives the expected state at time $t+1$ upon taking action a while in state s at time t.

2. The dependence of the state s_t on the policy π_θ plays a crucial role in their results. To express this dependence, they define:

$$\begin{aligned} h_1(s_0, \theta) &= f_0(s_0, \pi_\theta(s_0)) \\ h_t(s_0, \theta) &= f_{t-1}(h_{t-1}(s_0, \theta)) \end{aligned}$$

3. i.e.: $h_t(s_0, \theta)$ is equal to the state at time t when using policy π_θ and starting in state s_0 at time 0.

4. For an approximate model $\hat{T}$, they similarly define $\hat{h}_t$ in terms of the approximate transition functions $\hat{f}_t$.

5. Let $s_0, s_1, \ldots, s_H$ be the real-life state sequence obtained when executing the current policy π_θ. Then the true policy gradient is given by:

$$\nabla_\theta U(\theta) = \sum_{t=0}^{H} \nabla_s R(s_t) \left. \frac{dh_t}{d\theta} \right|_{s_0, s_1, \ldots, s_{t-1}}$$

6. Let $s_0, \hat{s}_1, \ldots, \hat{s}_H$ be the state sequence according to the model $\hat{T}$ when executing the policy. Then according to the model, the policy gradient is given by:

$$\sum_{t=0}^{H} \nabla_s R(\hat{s}_t) \left. \frac{d\hat{h}_t}{d\theta} \right|_{s_0, \hat{s}_1, \ldots, \hat{s}_{t-1}}$$

7. Two source of errors make the gradient estimate differ from the true policy gradient: the derivative $\frac{d\hat{h}_t}{d\theta}$ is an approximation of $\frac{dh_t}{d\theta}$.

8. The derivatives appearing in $\nabla_s R(\hat{s}_t)$ and $\frac{d\hat{h}_t}{d\theta}$ are evaluated along the wrong trajectory (the estimated rather than the true trajectory).

9. Thus, what we can do is to evaluate the estimated gradient using a true trajectory.

$$\nabla_\theta \hat{U}(\theta) = \sum_{t=0}^{H} \nabla_s R(s_t) \left. \frac{d\hat{h}_t}{d\theta} \right|_{s_0, s_1, \ldots, s_{t-1}}$$

10. The paper also provides convergence result.

4 Questions

1. How is the model parameterized in this paper?

2. In this approach, we need to run the policy in the real MDP every time we want to make an update to the policy. Why do we expect this to be better than just model-free RL?