DRIVE: Distributional Model-based Reinforcement Learning via Variational Inference

1 From standard RL to distributional perspective

1.1 Objective

$$\max_\pi V^\pi(s) = \mathbb{E}_\pi[Q^\pi(s,a)], \forall s \in \mathcal{S} \tag{1}$$

$$\max_\pi \log p^\pi(\mathcal{O}=1|s) = \log \mathbb{E}_\pi[p^\pi(\mathcal{O}=1|s,a)], \forall s \in \mathcal{S}, \tag{2}$$

where $\mathcal{O}$ is a binary random variable indicating the optimility when $\mathcal{O}=1$.

1.2 Distributional Bellman operator

$$\begin{aligned}
\mathcal{T}^\pi \overbrace{U(s,a)}^{p(U|s,a)} &\overset{\text{D}}{:=} \overbrace{r(s,a)+\gamma U(s',a')}^{q(U|s,a)} \qquad s' \sim P(\cdot|s,a), a' \sim \pi(\cdot|s') \\
(\mathcal{T}^\pi)^H U(s_t,a_t) &\overset{\text{D}}{:=} r_{<H} + \gamma^H U(s_{t+H}, a_{t+H}) \qquad \tau \sim \pi,\ r_{<H} := \sum_{n=0}^{H-1} \gamma^n r(s_{t+n}, a_{t+n})
\end{aligned} \tag{3}$$

where the equality is held under probability laws.

2 Algorithm

2.1 Variational Bound

Considering $p(\mathcal{O}=1|U,s,a) \propto \exp(U)$,

$$\begin{aligned}
\log p_\psi^{\pi_\theta}(\mathcal{O}=1|s) &\geq \mathcal{L}(\theta,\phi,\psi;s) \\
&= -D_{\text{KL}}(\underbrace{q_\phi(a|\mathcal{O}=1,s)}_{\text{posterior}}||\pi_\theta(a|s)) + \mathbb{E}_{q_\phi(a|\mathcal{O}=1,s)}[\log \underbrace{p_\psi(\mathcal{O}=1|s,a)}_{\text{optimility distribution}}] \\
&= -D_{\text{KL}}(q_\phi(a|\mathcal{O}=1,s)||\pi_\theta(a|s)) + \mathbb{E}_{q_\phi(a|\mathcal{O}=1,s)}[\log \int \underbrace{p(\mathcal{O}=1|U,s,a)}_{\propto \exp(U)} p_\psi(U|s,a)dU] \\
&\geq -\underbrace{D_{\text{KL}}(q_\phi(a|\mathcal{O}=1,s)||\pi_\theta(a|s))}_{\text{complexity}} + \underbrace{\mathbb{E}_{q_\phi(a|\mathcal{O}=1,s),q(U|s,a)}[U|s,a]}_{\text{reparametrized PG}} - \mathbb{E}_{q_\phi(a|\mathcal{O}=1,s)}\underbrace{[D_{\text{KL}}(q(U|s,a)||p_\psi(U|s,a))]}_{\text{regularizer \& policy evaluation}} - \text{const}
\end{aligned}$$

2.2 Updating rule

$$\texttt{Value:}\ \mathcal{J}(\psi) = \mathbb{E}_{q(U|s,a)}[-\log p_\psi(U|s,a)] \tag{4}$$

$$\texttt{Policy:}\ \cancel{\mathcal{J}(\theta) = D_{\text{KL}}(\pi_\theta||q_\phi)} \tag{5}$$

$$\texttt{Posterior + Policy:}\ \mathcal{J}(\theta,\phi) = -\underbrace{\mathbb{E}_{q_\phi(a|\mathcal{O}=1,s),q(U|s,a)}[U|s,a]}_{\mathcal{J}_U} + \underbrace{D_{\text{KL}}(q_\phi(a|\mathcal{O}=1,s)||\pi_\theta(a|s))}_{\mathcal{J}_{\text{KL}}^{(1)}} + \underbrace{\mathbb{E}_{q_\phi(a|\mathcal{O}=1,s)}[D_{\text{KL}}(q(U|s,a)||p_\psi(U|s,a))]}_{\mathcal{J}_{\text{KL}}^{(2)}} \tag{6}$$

2.3 Approximation

Denote the learned transition model as $\hat{f}$,

For $\mathcal{J}_U$

$$\mathcal{J}_U = \mathbb{E}_{q_\phi, \pi_\theta, \hat{f}, p_\psi(U|s_{t+H}, a_{t+H})} \left[r_{<H} + \gamma^H U(s_{t+H}, a_{t+H}) \right]$$

For $\mathcal{J}_{\mathbf{KL}}^{(2)}$

$$\begin{aligned} q(U|s,a) &= \frac{1}{\gamma} \mathbb{E}_{\pi_\theta, \hat{f}} \left[p_\psi\left(\frac{U - r_{<H}}{\gamma}\right) \right] \\ &= \mathbb{E}_{\pi_\theta, \hat{f}} \left[\mathcal{N}(r_{<H} + \gamma^H \mu(U_{t+H}), \gamma^{2H} \sigma^2(U_{t+H})) \right], \qquad \text{if } p_\psi \sim \mathcal{N}(\mu, \sigma^2) \\ &\approxeq \frac{1}{N} \sum_i^N \mathcal{N}(r_{<H}(\tau_i) + \gamma^H \mu_i(U_{t+H}), \gamma^{2H} \sigma_i^2(U_{t+H})) \end{aligned}$$

- Typically $N = 1$ works well
- When $N > 1$, it is more likely exploiting model error.

Use reparametrization trick to make above all differentiable w.r.t. both θ, ϕ.

Algorithm 1 DRIVE

```
while not converged do
  for each update step j = 1, ..., C do
    model learning
    posterior + actor learning Eq. 6
    value learning Eq. 4
  end for
  for each environment step j = 1, ..., T do
    collect data
    add data to replay buffer D
  end for
end while
```