RL as an Inference Problem

Zhu Mingren

April 7, 2021

Contents

Contents

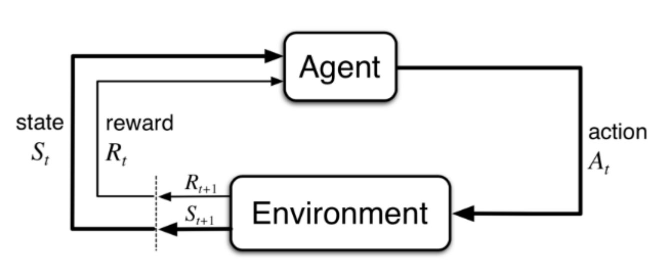
Markov Decision Process (MDP)

Basic Elements

- Environment with a set of states $\mathcal{S}$
- Agent with a set of possible actions $\mathcal{A}$
- Environment dynamics: $p(s_{t+1}|s_t, a_t)$
- Reward function: $r(s, a)$
- Policy: $\pi(a|s)$
- Trajectory of an agent: $\tau = (s_1, a_1, r_1, s_2, a_2, r_2, \dots)$

Markov Decision Process (MDP)

Interactions



The interactions between an agent and the environment.

Markov Decision Process (MDP)

Returns and Value Functions

- Return (cumulative reward) starting step t:
  $G_t = r_{t+1} + r_{t+2} + \cdots + r_T = r_{t+1} + G_{t+1}$
- Return with discount $0 < \gamma \leq 1$:
  $G_t = r_{t+1} + \gamma r_{t+2} + \cdots = \sum_{k=0} \gamma^k r_{t+1+k} = r_{t+1} + \gamma G_{t+1}$
- Value function of a state s:
  $V_\pi(s) = \mathbb{E}_\pi\{G_t | s_t = s\} = \mathbb{E}_\pi\{\sum_{k=0} \gamma^k r_{r+1+k} | s_t = s\}$
- Value function of the state-action pair (s, a):
  $Q_\pi(s, a) = \mathbb{E}_\pi\{G_t | s_t = s, a_t = a\} = \mathbb{E}_\pi\{\sum_{k=0} \gamma^k r_{t+1+k} | s_t = a, a_t = a\}$

Markov Decision Process (MDP)

Bellman Equations

- Bellman equation for $V_\pi(s)$:

$$V_\pi(s) = \sum_a \pi(a|s) \sum_{s'} p(s'|s,a)[r(s,a) + \gamma V_\pi(s')]$$

- Bellman equation for $Q_\pi(s,a)$:

$$Q_\pi(s,a) = r(s,a) + \gamma \sum_{s'} p(s'|s,a) \sum_{a'} \pi(a'|s') Q_\pi(s',a')$$

The Q-learning Algorithm

Optimal Policy and Value Functions

Given the optimal policy π^*, the optimal value functions are $V^*(s)$ and $Q^*(s, a)$, where the "optimal" means there is no other policy can make the value functions greater than these two.

$$V^*(s) \geq V_\pi(s), \quad \forall s \in \mathcal{S}, \quad \forall \pi$$
$$Q^*(s, a) \geq Q_\pi(s), \quad \forall (s, a) \in \mathcal{S} \times \mathcal{A}, \quad \forall \pi$$

π^* can be derived by: $\pi^*(a|s) = \delta(a = arg \max_a Q^*(s, a))$
V^* can be derived by: $V^*(s) = \max_a Q^*(s, a)$

The Q-learning Algorithm

Q-learning (Table Q)

The bellman equation for $Q^*(s, a)$:

$$Q^*(s, a) = \mathbb{E}_{s' \sim p(s'|s,a)}\{r(s, a) + \gamma \max_{a'} Q^*(s', a')\}$$

It can be used to update $Q_\pi(s, a)$ even though the policy is not optimal:

$$Q_{new}(s, a) = \mathbb{E}_{s' \sim p(s'|s,a)}\{r(s, a) + \gamma \max_{a'} Q_{old}(s', a')\}$$

This updating can lead Q to be greater and going to converge:

$$Q_{old}(s, a) \leq Q_{new}(s, a) \rightarrow Q^*(s, a)$$

The Q-learning Algorithm

Deep Q-learning (DQN)

If the state s is in a high dimensional space such as pictures, the Q function can be parametrized to a neural network $Q_\theta(s, a)$. Then calculating the *targetQ* value:

$$targetQ = \mathbb{E}_{s' \sim p(s'|s,a)}\{r(s, a) + \gamma \max_{a'} Q_\theta(s', a')\}$$

Finally using GD to minimize the MSE loss:

$$\theta \leftarrow \theta - \epsilon \nabla_\theta ||targetQ - Q_\theta(s, a)||_2^2$$

DQN is not guaranteed to converge, but it works well in practice.

Contents

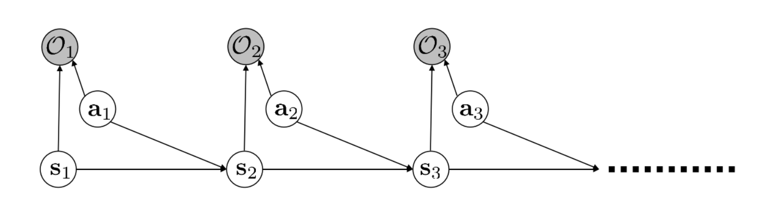
Background

Some Issues of RL

- – Can RL be used to explain human behavior?
  – Yes, the *inverse* RL.
- – Does RL provide a reasonable model of human behavior?
  – No, the data suited for RL needs to be optimal, but human behavior is non-optimal and stochastic.
- – Is there a better explanation of human behavior?
  – Yes, good behavior is still the most likely, and can be modeled in a probabilistic graph (PGM).

The PGM of Decision Making

The Probabilistic Graphical Model



The system dynamic is still $p(s'|s,a)$, and it should not change no matter how the agent behaves. There are some new binary variables notated as $\mathcal{O}_t$, $p(\mathcal{O}=1|s,a) \propto \exp\{r(s,a)\}$ means the probability of action a is optimal for state s. $r(s,a)$ plays a role similar to "reward function".

"Optimal" Behavior Sequence

The "optimal" behavior sequence here is a stochastic process with distribution $p(\tau = (s_t, a_t)_{t=1}^{T} | \mathcal{O}_{1:T} = 1)$, which can be derived by the bayes theorem:

$$p(\tau | \mathcal{O}_{1:T}) \propto p(\tau) p(\mathcal{O}_{1:T} | \tau) = p(\tau) \prod_{t=1}^{T} p(\mathcal{O}_t = 1 | s_t, a_t)$$
$$\propto [p(s_1) \prod_{t=1}^{T} p(s_{t+1} | s_t, a_t)] \exp\{\sum_{i=1}^{T} r(s_t, a_t)\}$$

It can be used to model suboptimal and stochastic behavior.

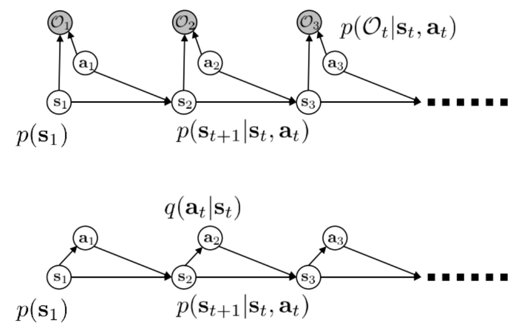
Control via Variational Inference

Constraint and Object

- The posterior $p(a_t|s_t, \mathcal{O}_{1:T})$ can be used as the optimal policy only when the system dynamic is $p(s_{t+1}|s_t, a_t, \mathcal{O}_{t:T})$.
- However, as it is said before, the system dynamic should not change no matter how the agent behaves, obviously $p(s_{t+1}|s_t, a_t, \mathcal{O}_{t:T}) \neq p(s_{t+1}|s_t, a_t)$ in an ordinary way.
- Even so, it can still be used as a policy, just not the optimal one under $p(s'|s, a)$, because it can not generate the "optimal" behavior sequence as $p(\tau|\mathcal{O}_{1:T})$.
- Then how to get the optimal policy under $p(s'|s, a)$? What policy can generate the same "optimal" behavior sequence as $p(\tau|\mathcal{O}_{1:T})$?

Control via Variational Inference

Variational Inference



Using policy $q(a|s)$ to approximate $p(\tau|\mathcal{O}_{1:T})$ under $p(s'|s, a)$.

Control via Variational Inference

Variational Inference

$$
\begin{aligned}
q(\tau) = p(s_1)\prod_{t=1}^{T} q(a_t|s_t)p(s_{t+1}|s_t, a_t) &\rightarrow p(\tau|\mathcal{O}_{1:T}) \\
\max \mathbb{ELBO}\{q||p\} &= \mathbb{E}_{\tau\sim q(\tau)}\{\log p(\tau, \mathcal{O}_{1:T}) - \log q(\tau)\} \\
&= \mathbb{E}_{\tau\sim q(\tau)}\{\sum_{t=1}^{T}[r(s_t, a_t) - \log q(a_t|s_t)]\} \\
&= \sum_{t=1}^{T}\mathbb{E}_{(s_t,a_t)\sim q}\{r(s_t, a_t)\} - \log q(a_t|s_t)\}
\end{aligned}
$$

Control via Variational Inference

Solve for $q(a_T|s_T)$

$$
\begin{aligned}
q(a_T|s_T) &= arg\max \mathbb{E}_{(s_T,a_T)\sim q}\{r(s_T,a_T) - \log q(a_t|s_t)\} \\
&\propto \exp\{r(s_T,a_T)\}
\end{aligned}
$$

Let $Q(s_T,a_T) = r(s_T,a_T)$ and $V(s_T) = \log\sum_a \exp\{r(s_T,a)\}$:

$$
q(a_T|s_T) = \exp\{Q(s_T,a_T) - V(s_T)\}
$$

$$
\mathbb{E}_{(s_T,a_T)\sim q}\{r(s_T,a_T) - \log q(a_t|s_t)\} = \mathbb{E}_{s_T\sim p(s_T|s_{T-1},a_{T-1})}\{V(s_T)\}
$$

Control via Variational Inference

Solve for $q(s_t|a_t)$

Let $Q(s_t, a_t) = r(s_t, a_t) + \mathbb{E}_{s_{t+1} \sim p(s_{t+1}|s_t, a_t)}\{V(s_{t+1})\}$:

$$
\begin{aligned}
q(a_t|s_t) &= \mathit{arg}\max \mathbb{E}_{(s_t, a_t) \sim q}\{Q(s_t, a_t) - \log q(a_t|s_t)\} \\
&\propto \exp\{Q(s_t, a_t)\}
\end{aligned}
$$

Let $V(s_t) = \log \sum_a \exp\{Q(s_t, a)\}$:

$$q(a_t|s_t) = \exp\{Q(s_t, a_t) - V(s_t)\}$$
$$\mathbb{E}_{(s_t, a_t) \sim q}\{Q(s_t, a_t) - \log q(a_t|s_t)\} = \mathbb{E}_{s_t \sim p(s_t|s_{t-1}, a_{t-1})}\{V(s_t)\}$$

Control via Variational Inference

Standard Q-learning

$$Q(s,a) = r(s,a) + \mathbb{E}_{s' \sim p(s'|s,a)}\{V(s')\}$$
$$V(s) = \max_a Q(s,a)$$
$$\pi(a|s) = \delta(a = arg\max_{a'} Q(s,a'))$$

Soft Q-learning

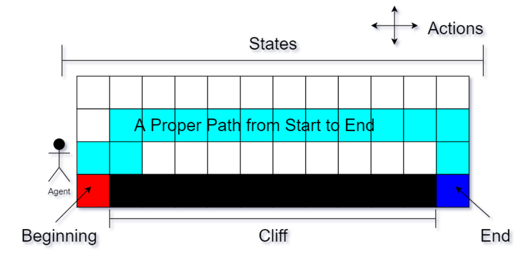
$$Q(s,a) = r(s,a) + \mathbb{E}_{s' \sim p(s'|s,a)}\{V(s')\}$$
$$V(s) = \log \sum_a \exp\{Q(s,a)\}$$
$$\pi(a|s) = \exp\{Q(s,a) - V(s)\}$$

$$r(s,a) \nearrow \quad \Longrightarrow \quad \text{Soft-Q} \rightarrow \text{Standard-Q}$$

Contents

Toy Example

Cliff Walking



The goal of the agent is to walk from the beginning to the end, as short a path as possible and to avoid falling off the cliff (or returning to the beginning).

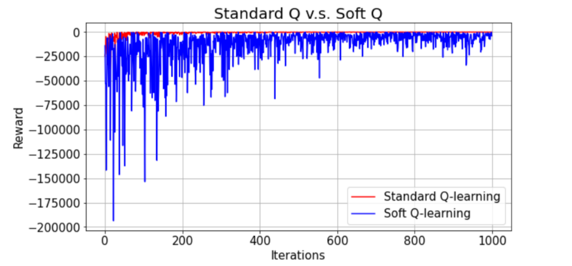
Toy Example

Settings

Reward		Learning		ϵ-greedy	
normal step	-1	α	0.8	ϵ_{max}	0.9
falling off the cliff	-100	γ	0.95	ϵ_{min}	0.1
arriving the end	0	episodes	1000	descent ratio	0.001

Toy Example

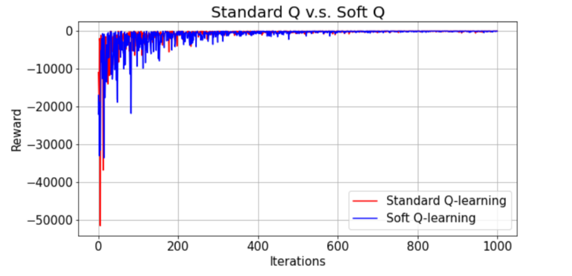
Result A



Soft Q-learning is not as good as standard Q-learning in the case because the dynamic of the system is deterministic with only one optimal policy.

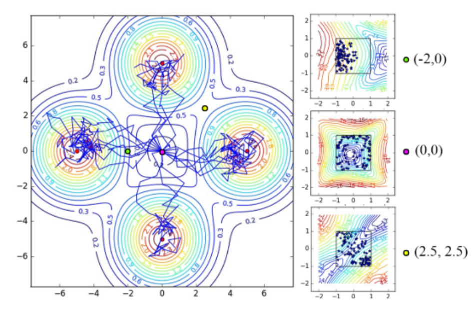
Toy Example

Result B



If rewards are magnified 10 times during training with Soft-Q, it turns out that Soft-Q can be very similar to the standard Q-learning.

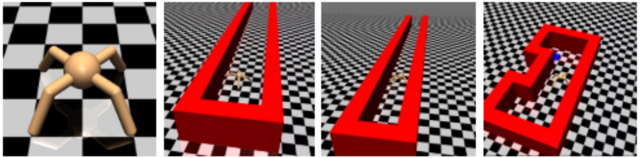
Didactic Example: Multi-Goal Environment



Soft-Q can be used to explore multi-goal/multi-mode environments.

Experiments from Soft-Q Paper

Accelerating Training with Pretrained Policies



(a) (b) (c) (d)

Soft-Q can be used to generate pre-trained policies.

Contents

Summary

Benefits of Soft Optimality

- Improves exploration and prevents entropy collapse
- Empirically, policies are easier to fine-tune for more specific tasks
- Better robustness (due to wider coverage of states)
- Reduces to hard optimality (by increasing the magnitude of the rewards)
- Good model for human behavior

References

- Reinforcement Learning & Control Through Inference in GM, Maruan Al-Shedivat, CMU
- Reframing Control as an Inference Problem, Sergey Levine, UC Berkeley
- Reinforcement Learning with Deep Energy-Based Policies, Tuomas Haarnoja, 2017
- Reinforcement Learning and Control as Probabilistic Inference: Tutorial and Review, Sergey Levine, 2018

Summary

Thanks for your attention!