Maximum Entropy Reinforcement Learning

Guo-Yu Yang

2020.10.26
Outline

Background

Redefinition: Bellman function

Some algorithms

Summary

Reference
Background: entropy regularized policy optimization problem

- Given K actions and the corresponding reward vector $\mathbf{q} \in \mathbb{R}^K$, and $\pi$ is a probability distribution in action space. The entropy regularized policy optimization problem:

$$\max_{\pi} \{\pi \mathbf{q} + \tau \mathcal{H}(\pi)\}$$

$\tau \geq 0$ controls the degree of exploration.

- Entropy of policy:

$$\mathcal{H}(\pi) = -\sum_{\pi_a \in \pi} \pi_a \log(\pi_a)$$

The entropy of deterministic policy is relatively low, and the entropy of random policy is relatively high.
Background: Softmax function

- The Softmax $\mathcal{F}_\tau$ function:
  \[ \mathcal{F}_\tau(q) = \tau \log \sum_a e^{q_a / \tau} \]

- And we define the Soft-Indmax $f_\tau$ function:
  \[ f_\tau(q) = \nabla \mathcal{F}_\tau(q) = \frac{e^{q / \tau}}{\sum_a e^{q_a / \tau}} = \frac{e^{(q - \mathcal{F}_\tau(q)) / \tau}}{\sum_a e^{q_a / \tau}} \]

- Soft-Indmax function gets the confidence of each action (different from Hardmax function), and it’s possible to explore.
The connection between entropy regularized policy optimization problem and Softmax function:

\[ \mathcal{F}_\tau(q) = \max_{\pi} \{ \pi q + \tau \mathcal{H}(\pi) \} = f_\tau(q) \cdot q + \tau \mathcal{H}(f_\tau(q)) \]

where \( \pi^* = f_\tau(q) \) and \( \mathcal{F}_\tau(q) = q_a - \tau \log \pi_a, \forall a. \)

- The Softmax value is the upper bound on the maximum value, and the gap between them is the entropy of the policy.

- When \( \tau \to 0 \), The entropy regularized policy optimization problem becomes the standard expected reward objective, where the optimal solution is the hard-max policy.
Proof:

The first equation: Let $\mathcal{F}_\tau^*$ denotes as conjugate of $\mathcal{F}_\tau$:

$$\mathcal{F}_\tau^*(p) = \sup_q \{ p \cdot q - \mathcal{F}_\tau(q) \} = \tau p \log p$$

For $\sum_{p \in \mathbb{P}} p = 1$. Since $\mathcal{F}_\tau$ is closed and convex, $\mathcal{F}_\tau = \mathcal{F}_{\tau^{**}}$:

$$\mathcal{F}_\tau(q) = \sup_p \{ p \cdot q - \tau p \log p \}$$

The second equation uses Lagrange multiplier method:

$$L = \pi (q - \tau \log \pi) + \lambda (1 - 1 \cdot \pi)$$

KKT condition:

$$1 - 1 \cdot \pi = 0 \ ; \ \tau \log \pi = q - v \ (v = \lambda + \tau)$$
Redefinition: Bellman function

- **Standard reinforcement learning objective:**
  \[ \pi^*_{std} = \arg \max_{\pi} \sum_{t=0}^{\infty} \mathbb{E}_{\rho_\pi} [\gamma^t r_t] \]

- **Maximum entropy reinforcement learning objective:**
  \[ \pi^*_{MaxEnt} = \arg \max_{\pi} \sum_{t=0}^{\infty} \mathbb{E}_{\rho_\pi} [\gamma^t (r_t + \tau H(\pi))] \]

- **Original Q-function:**
  \[ Q^*(s_t, a_t) = r_t + \mathbb{E}_{\rho_{\pi^*}} [\sum_{l=1}^{\infty} \gamma^l r_{t+l}] \]

- **Soft Q-function:**
  \[ Q^*_{soft}(s_t, a_t) = r_t + \mathbb{E}_{\rho_{\pi^*}} [\sum_{l=1}^{\infty} \gamma^l (r_{t+l} + \tau H(\pi^*))] \]
Redefinition: Bellman function

- Original optimal value function:
  \[ V^*(s) = \max_{a \in A} Q^*(s, a) \]

- Soft-value function:
  \[ V_{\text{soft}}^*(s) = \tau \log \int_A \exp\left(\frac{1}{\tau} Q_{\text{soft}}^*(s, a')\right) da' \]

- According to the above re-definition, we have following conclusions:
  1. Maximum entropy policy given by (Boltzmann distribution):
     \[ \pi_{\text{MaxEnt}}^*(a_t|s_t) = \exp\left(\frac{1}{\tau} (Q_{\text{soft}}^*(s_t, a_t) - V_{\text{soft}}^*(s_t))\right) \]
  2. The soft Bellman Equation:
     \[ Q_{\text{soft}}^*(s_t, a_t) = r(s_t, a_t) + \gamma \mathbb{E}_{\pi^*}[V_{\text{soft}}^*(s_{t+1})] \]
Redefinition: policy improvement theorem

- Policy improvement theorem: Given a policy $\pi$, define a new policy $\tilde{\pi}$ as

$$\tilde{\pi}(\cdot|s) \propto \exp(Q_{\text{soft}}^\pi(s, \cdot))$$

Assume that $Q$ is bounded and $\int \exp(Q(s, a)) da$ is bounded for any $s$. Then $Q_{\tilde{\pi}}(s, a) \geq Q^\pi(s, a), \forall s, a$.

- Proof:
  - Now, we proof the next inequality:

$$\tau \mathcal{H}(\pi(\cdot|s)) + \mathbb{E}_{a \sim \pi}[Q_{\text{soft}}^\pi(s, a)] \leq \tau \mathcal{H}(\tilde{\pi}(\cdot|s)) + \mathbb{E}_{a \sim \tilde{\pi}}[Q_{\text{soft}}^\pi(s, a)]$$

According to the principle of normalization:

$$\tilde{\pi}(\cdot|s) = \exp\left(\frac{1}{\tau}(Q_{\text{soft}}^\pi(s, \cdot) - V_{\text{soft}}^\pi(s))\right)$$

$$Q_{\text{soft}}^\pi(s, \cdot) = V_{\text{soft}}^\pi(s) + \tau \log(\tilde{\pi}(\cdot|s))$$

Redefinition: policy improvement theorem

- Proof:
  - Therefore, we have the following equation:
    \[
    \tau \mathcal{H}(\pi(\cdot|s)) + \mathbb{E}_{a \sim \pi}[Q_{\text{soft}}^\pi(s, a)] = \tau \mathcal{H}(\pi(\cdot|s)) + V_{\text{soft}}^\pi(s) + \tau \pi(\cdot|s) \log(\bar{\pi}(\cdot|s))
    \]
    \[
    = -\tau D_{KL}(\pi(\cdot|s)\|\bar{\pi}(\cdot|s)) + V_{\text{soft}}^\pi(s)
    \]
    \[
    \leq \tau \mathcal{H}(\bar{\pi}(\cdot|s)) + V_{\text{soft}}^\pi(s) + \tau \bar{\pi}(\cdot|s) \log(\bar{\pi}(\cdot|s))
    \]
    \[
    = \tau \mathcal{H}(\bar{\pi}(\cdot|s)) + \mathbb{E}_{a \sim \bar{\pi}}[Q_{\text{soft}}^\pi(s, a)]
    \]
  - Then we can show that:
    \[
    Q_{\text{soft}}^\pi(s, a) = \mathbb{E}_{s_1}[r_0 + \gamma(\mathcal{H}(\pi(\cdot|s_1)) + \mathbb{E}_{a_1 \sim \pi}[Q_{\text{soft}}^\pi(s_1, a_1)])]
    \]
    \[
    \leq \mathbb{E}_{s_1}[r_0 + \gamma(\mathcal{H}(\bar{\pi}(\cdot|s_1)) + \mathbb{E}_{a_1 \sim \bar{\pi}}[Q_{\text{soft}}^\pi(s_1, a_1)])]
    \]
    \[
    = \mathbb{E}_{s_1}[r_0 + \gamma(\mathcal{H}(\bar{\pi}(\cdot|s_1)) + r_1)] + \gamma^2 \mathbb{E}_{s_2}[\mathcal{H}(\pi(\cdot|s_2)) + \mathbb{E}_{a_2 \sim \pi}[Q_{\text{soft}}^\pi(s_2, a_2)]]
    \]
    \[
    \ldots
    \]
    \[
    \leq \mathbb{E}_{\rho \sim \bar{\pi}}[r_0 + \sum_{t=1}^{\infty} \gamma^t(\mathcal{H}(\bar{\pi}(\cdot|s_t)) + r_t)]
    \]
    \[
    = Q_{\bar{\pi}}^\pi(s, a)
    \]
Redefinition: Bellman function

- According to policy improvement theorem, after several iterations we can get:
  \[ \pi^*_{MaxEnt}(a_t|s_t) = \exp\left(\frac{1}{\tau}(Q^*_{soft}(s_t, a_t) - V^*_{soft}(s_t))\right) \]

- Proof of soft Bellman Equation:
  - Soft Q-function rewritten as:
    \[ Q^\pi(s, a) = r(s, a) + \gamma \mathbb{E}_{s' \sim p}[\tau H(\pi(\cdot|s')) + \mathbb{E}_{a' \sim \pi(\cdot|s')} (Q^\pi(s', a'))] \]
    - according to policy improvement theorem:
      \[ Q^\pi(s, a) = V^\pi(s) + \tau \log(\pi(a|s)) \]
    - Combine the two equation:
      \[ Q^\pi(s, a) = r(s, a) + \gamma \mathbb{E}_{s' \sim p}[V^\pi(s')] \]
Algorithm: Soft Q-Iteration

- Let $Q$ and $V$ be bounded. The fixed-point iteration:

$$Q(s_t, a_t) \leftarrow r(s_t, a_t) + \gamma \mathbb{E}_{s_{t+1} \sim \rho}[V(s_{t+1})], \forall s_t, a_t$$

$$V(s_t) \leftarrow \tau \log \left( \int_A \exp \left( \frac{1}{\tau} Q(s_t, a') \right) da' \right), \forall s_t$$

converges to $Q^*$ and $V^*$ respectively.

- Disadvantage:
  1. The soft Bellman backup cannot be performed exactly in continuous or large state and action spaces.
  2. Sampling from the energy-based model in $\tilde{\pi}(\cdot|s) \propto \exp(Q_{soft}^\pi(s, \cdot))$ is intractable in general.
Algorithm: Soft Q-Iteration

Proof:

We show that the soft value iteration operator $T$, defined as:

$$TQ(s, a) = r(s, a) + \gamma \mathbb{E}_{s' \sim p_s}[\log \int \exp Q(s', a') da']$$

Define a norm $\|Q_1 - Q_2\| = \max_{s, a} |Q_1(s, a) - Q_2(s, a)|$. Suppose $\varepsilon = \|Q_1 - Q_2\|:

$$\log \int \exp(Q_1(s', a')) da' \leq \log \int \exp(Q_2(s', a') + \varepsilon) da'$$

$$= \log(\exp(\varepsilon) \int \exp Q_2(s', a') da')$$

$$= \varepsilon + \log \int \exp(Q_2(s', a')) da'$$

Therefore, $\|TQ_1 - TQ_2\| \leq \gamma \varepsilon = \gamma \|Q_1 - Q_2\|$. So $T$ is a contraction.
Algorithm: Soft Q-learning

In order to solve the above problem of Soft Q-iteration, we use stochastic optimization problem to model. The following is the pseudocode of Soft Q-learning:

```
Algorithm 1 Soft Q-learning
Input: \( \theta, \phi \)
1: \( \bar{\theta} \leftarrow \theta, \bar{\phi} \leftarrow \phi \)
2: \( \mathcal{D} \leftarrow \emptyset \)
3: for each epoch do
   4:     for each t do
   5:         \( a_t = f^\theta(\xi, s_t) \), where \( \xi \sim \mathcal{N}(0, I) \)
   6:         \( s_{t+1} \sim p(s_{t+1}|s_t, a_t) \)
   7:         \( \mathcal{D} \leftarrow \mathcal{D} \cup \{(s_t, a_t, r(s_t, a_t), s_{t+1})\} \)
   8:     Update soft Q-function parameters
   9:     Update policy
10:     end for
11: if epoch \% update_interval = 0 then
12:     \( \bar{\theta} \leftarrow \theta, \bar{\phi} \leftarrow \phi \)
13: end if
14: end for
15: return \( \theta, \phi \)
```

Algorithm: Soft Q-learning

- Firstly, we model the soft Q-function with a function approximator with parameters $\theta$, $Q_\theta(s_t, a_t)$. We will optimize the soft Bellman error $|TQ - Q|$ to find optimal Q-function.
- Update soft Q-learning parameters:
  1. Sample a minibatch from the replay memory: $\{s^{(i)}_t, a^{(i)}_t, r^{(i)}_t, s^{(i)}_{t+1}\}_{i=0}^N \sim D$
  2. Sample $M$ uniform actions: $\{a^{(i,j)}\}_{j=0}^M \sim q_{a'}$
  3. Compute empirical soft value with next equation:

$$V_\theta(s^{(i)}_{t+1}) = \tau \log \mathbb{E}_{q_{a'}} \left[ \frac{\exp(\frac{1}{\tau} Q_\theta(s_t, a'))}{q_{a'}(a')} \right]$$

   It use importance sample, where $q_{a'}$ is a non-zero distribution over the action space.
  4. Compute empirical gradient of next equation (equivalent to Soft Q-iteration):

$$g^{(i)}_\theta \leftarrow \nabla_\theta \left( \mathbb{E}_{s_t \sim q_s, a_t \sim q_a} \left[ \frac{1}{2} (Q_\theta(s_t, a_t) - (r(s_t, a_t) + \gamma \mathbb{E}_{s_{t+1} \sim p}[V_\theta(s_{t+1})]))^2 \right] \right)$$

  5. Update $\theta$: $\theta \leftarrow \text{ADAM}(\theta, g^{(i)}_\theta)$
Algorithm: Soft Q-learning

- Secondly, we want to learn a state-conditioned stochastic neural network \( a_t = f^\phi(\xi, s_t) \).
- Update policy process:
  1. Sample M latents for each state: \( \{\xi^{(i,j)}\}_{j=0}^M \sim \mathcal{N}(0, I) \)
  2. Evaluate actions: \( a_t = f^\phi(\xi^{(i,j)}, s_t^{(i)}) \)
  3. Evaluate empirical Stein variational gradient using next equation:
     \[
     \Delta f^\phi(\cdot; s_t) = \mathbb{E}_{a_t \sim \pi^\phi}[\kappa(a_t, f^\phi(\cdot; s_t))\nabla_{a'} Q_\theta(s_t, a')|_{a'=a_t} + \tau \nabla_{a'} \kappa(a', f^\phi(\cdot; s_t))|_{a'=a_t}]
     \]
  4. Compute empirical estimate \( g^{(i)}_\phi = \hat{\nabla}_{\phi} J_\pi \) of next equation:
     \[
     \frac{\partial J_\pi(\phi; s_t)}{\partial \phi} \propto \mathbb{E}_\xi \left[ \Delta f^\phi(\xi; s_t) \frac{\partial f^\phi(\xi, s_t)}{\partial \phi} \right]
     \]
     \[
     J_\pi(\phi; s_t) = D_{KL}(\pi^\phi(\cdot|s_t)||\exp(\frac{1}{\tau}(Q_\theta(s_t, \cdot) - V_\theta(s_t))))
     \]
  5. Update \( \phi \): \( \phi \leftarrow \text{ADAM}(\phi, g^{(i)}_\phi) \)
Algorithm: Soft Q-learning

- Experimental environment:

![Diagram showing different experimental environments](image)

- Experimental environment:

![Graphs showing performance comparison](image)
Algorithm: Maximum entropy monte-carlo planning

- UCT can only guarantee a polynomial convergence rate of finding the best action at the root in MCTS.
- MENTS (Maximum Entropy for Tree Search) improves UCT to exponential convergence speed:
  1. Using E2W (empirical exponential weight) as the tree policy.
  2. Evaluating each search node by softmax values back-propagated from simulations.
- The stochastic softmax bandit:
  The target is finding the policy with maximum softmax value in action set $A$:
  \[
  V^*_{soft} = \mathcal{F}_\tau(r)
  \]
  An important assumption: reward function of action, $r(a)$, are from $\sigma^2$-subgaussian.

Algorithm: Maximum entropy monte-carlo planning

- **Empirical exponential weight (E2W):** enforce enough exploration to guarantee good estimation of $q$, and make the policy converge to $\pi^*$ asymptotically. The algorithm selects an action by sampling from the distribution:

$$\pi_t(a) = (1 - \lambda_t)f_\tau(q)(a) + \lambda_t \frac{1}{|A|}$$

where $\lambda_t = \epsilon|A| / \log(t + 1)$ is a decay rate for exploration, with exploration parameter $\epsilon > 0$.

- **Theoretical guarantee of E2W:** In the softmax stochastic bandit problem:

$$\lim_{t \to \infty} t\mathcal{E}^t = \sigma^2 \tau^2 (\sum_a \exp(r(a)/\tau))^2$$

where $t$ is number of iterations, $\sigma$ is the assumption of the subsequent distribution, $\mathcal{E}_t$ is the mean square error in stochastic bandit problem.
Algorithm: Maximum entropy monte-carlo planning

- MENTS policy:

\[
\pi_t(a|s) = (1 - \lambda_s)f_\tau(Q_{soft}(s))(a) + \lambda_s \frac{1}{|A|}
\]

where \( \lambda_s = \epsilon |A|/ \log(\sum_a N(s, a) + 1) \). \( Q_{soft} \) denote a \( A \)-dimensional vector with components \( Q_{soft}(s, a) \). Q-values using the softmax backup:

\[
Q_{soft}(s_t, a_t) = \begin{cases} 
  r(s_t, a_t) + R & \text{if } t = T - 1 \\
  r(s_t, a_t) + \mathcal{F}_\tau(Q_{soft}(s_{t+1})) & \text{if } t < T - 1
\end{cases}
\]

\( R \) is the evaluation value by rollout policy. Finally, MENTS proposes the action with the maximum estimated softmax value.
Algorithm: Maximum entropy monte-carlo planning

- Theoretical guarantee:
  Let $a_t$ be the action returned by MENTS at iteration $t$. Then for large enough $t$ with some constant $C$:
  \[
P(a_t \neq a^*) \leq Ct \exp\left(-\frac{t}{(\log t)^3}\right)
  \]

- Experimental effect:

<table>
<thead>
<tr>
<th></th>
<th>BeamRider</th>
<th>Breakout</th>
<th>Q*bert</th>
<th>Seaquest</th>
<th>SpaceInvaders</th>
</tr>
</thead>
<tbody>
<tr>
<td>DQN</td>
<td>19280</td>
<td>345</td>
<td>14558</td>
<td>1142</td>
<td>625</td>
</tr>
<tr>
<td>UCT</td>
<td>21952</td>
<td>367</td>
<td>16010</td>
<td>1129</td>
<td>656</td>
</tr>
<tr>
<td>MENTS</td>
<td>18576</td>
<td>386</td>
<td>18336</td>
<td>1161</td>
<td>1503</td>
</tr>
</tbody>
</table>
Other Algorithm 1

1. Path Consistency Learning (PCL):

2. Prove the equivalence of soft Q-learning and policy gradients

3. Hierarchical maximum entropy reinforcement learning

4. Combine Soft Q-learning policies to produce better policy
Other Algorithm II

5. Soft Actor-Critic algorithm

6. Dynamically and automatically tuning the temperature parameter in Soft Actor-Critic Algorithm

7. Soft policy gradient method
Summary

1. Entropy regularized policy optimization problem: trade off exploration-exploitation directly
2. Bellman equation redefinition: combine with entropy regularized policy optimization problem and softmax function
3. Simple algorithm: soft Q-iteration
4. More practical algorithm: soft Q-learning
5. Maximum Entropy for Tree Search: variations of MCTS
Note: Papers in this area mainly come from UC Berkeley, U alberta.


Thankyou!