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 $\mathbf{q} \in \mathbb{E}^K$, and π 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:

$$\mathcal{F}_\tau(\mathbf{q}) = \tau \log \sum_a e^{q_a/\tau}$$

- And we define the Soft-Indmax $\mathbf{f}_\tau$ function:

$$\mathbf{f}_\tau(\mathbf{q}) = \nabla \mathcal{F}_\tau(\mathbf{q}) = \frac{e^{\boldsymbol{q}/\tau}}{\sum_a e^{q_a/\tau}} = e^{(\boldsymbol{q}-\mathcal{F}_\tau(\boldsymbol{q}))/\tau}$$

- Soft-Indmax function gets the confidence of each action (different from Hardmax function), and it's possible to explore.

Background: the connection

- The connection between entropy regularized policy optimization problem and Softmax function:

$$\mathcal{F}_\tau(\mathbf{q}) = \max_\pi\{\pi\mathbf{q} + \tau\mathcal{H}(\pi)\} = \mathbf{f}_\tau(\mathbf{q})\cdot\mathbf{q} + \tau\mathcal{H}(\mathbf{f}_\tau(\mathbf{q}))$$

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

Background: the connection

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

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

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

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

  - The second equation uses Lagrange multiplier method:

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

    KKT condition:

$$1 - \mathbf{1}\cdot\pi = 0\ ; \tau\log\pi = \mathbf{q} - v\ (v = \lambda + \tau)$$

Redefinition: Bellman function

- Standard reinforcement learning objective:

$$\pi^*_{std} = \arg\max_{\pi} \sum_{t=0} \mathbb{E}_{\rho_\pi}[\gamma^t r_t]$$

- Maximum entropy reinforcement learning objective:

$$\pi^*_{MaxEnt} = \arg\max_{\pi} \sum_{t=0} \mathbb{E}_{\rho_\pi}[\gamma^t (r_t + \tau\mathcal{H}(\pi))]$$

- 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:

$$Q^*_{soft}(s_t, a_t) = r_t + \mathbb{E}_{\rho_{\pi^*}}[\sum_{l=1}^{\infty} \gamma^l (r_{t+l} + \tau\mathcal{H}(\pi^*))]$$

Redefinition: Bellman function

- Original optimal value function:

$$V^*(s) = \max_{a \in \mathcal{A}} Q^*(s, a)$$

- Soft-value function:

$$V^*_{soft}(s) = \tau \log \int_{\mathcal{A}} \exp(\frac{1}{\tau} Q^*_{soft}(s, \mathbf{a}'))\, d\mathbf{a}'$$

- According to the above re-definition, we have following conclusions:
  1. Maximum entropy policy given by (Boltzmann distribution):

$$\pi^*_{MaxEnt}(a_t|s_t) = \exp\left(\frac{1}{\tau}(Q^*_{soft}(s_t, a_t) - V^*_{soft}(s_t))\right)$$

  2. The soft Bellman Equation:

$$Q^*_{soft}(s_t, a_t) = r(s_t, a_t) + \gamma \mathbb{E}_{\pi^*}[V^*_{soft}(s_{t+1})]$$

Redefinition: policy improvement theorem

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

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

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:

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

According to the principle of normalization:

$$\tilde{\pi}(\cdot|s) = \exp\left(\frac{1}{\tau}(Q^{\pi}_{soft}(s,\cdot) - V^{\pi}_{soft}(s))\right)$$

$$Q^{\pi}_{soft}(s,\cdot) = V^{\pi}_{soft}(s) + \tau\log(\tilde{\pi}(\cdot|s))$$

Tuomas Haarnoja et al. "Reinforcement Learning with Deep Energy-Based Policies". In:Proceedings of the 34th International Conference on Machine Learning, ICML 2017.

Redefinition: policy improvement theorem

- Proof:
  - Therefore, we have the following equation:

$$
\begin{aligned}
\tau\mathcal{H}(\pi(\cdot|s)) + \mathbb{E}_{a\sim\pi}[Q^{\pi}_{soft}(s,a)] &= \tau\mathcal{H}(\pi(\cdot|s)) + V^{\pi}_{soft}(s) + \tau\pi(\cdot|s)\log(\tilde{\pi}(\cdot|s)) \\
&= -\tau D_{KL}(\pi(\cdot|s)\|\tilde{\pi}(\cdot|s)) + V^{\pi}_{soft}(s) \\
&\le \tau\mathcal{H}(\tilde{\pi}(\cdot|s)) + V^{\pi}_{soft}(s) + \tau\tilde{\pi}(\cdot|s)\log(\tilde{\pi}(\cdot|s)) \\
&= \tau\mathcal{H}(\tilde{\pi}(\cdot|s)) + \mathbb{E}_{a\sim\tilde{\pi}}[Q^{\pi}_{soft}(s,a)]
\end{aligned}
$$

  - Then we can show that:

$$
\begin{aligned}
Q^{\pi}_{soft}(s,a) &= \mathbb{E}_{s_1}[r_0 + \gamma(\mathcal{H}(\pi(\cdot|s_1)) + \mathbb{E}_{a_1\sim\pi}[Q^{\pi}_{soft}(s_1,a_1)])] \\
&\le \mathbb{E}_{s_1}[r_0 + \gamma(\mathcal{H}(\tilde{\pi}(\cdot|s_1)) + \mathbb{E}_{a_1\sim\tilde{\pi}}[Q^{\pi}_{soft}(s_1,a_1)])] \\
&= \mathbb{E}_{s_1}[r_0 + \gamma(\mathcal{H}(\tilde{\pi}(\cdot|s_1)) + r_1)] + \gamma^2\mathbb{E}_{s_2}[\mathcal{H}(\pi(\cdot|s_2)) + \mathbb{E}_{a_2\sim\pi}[Q^{\pi}_{soft}(s_2,a_2)]] \\
&\cdots \\
&\le \mathbb{E}_{\rho\sim\tilde{\pi}}[r_0 + \sum_{t=1}\gamma^t(\mathcal{H}(\tilde{\pi}(\cdot|s_t)) + r_t)] \\
&= Q^{\tilde{\pi}}(s,a)
\end{aligned}
$$

Redefinition: Bellman function

- According to policy improvement theorem, after several iterations we can get:

$$\pi^*_{MaxEnt}(a_t|s_t) = \exp(\frac{1}{\tau}(Q^*_{soft}(s_t, a_t) - V^*_{soft}(s_t)))$$

- Proof of soft Bellman Equation:
  - Soft Q-function rewritten as:

$$Q^\pi(s, a) = r(s, a) + \gamma \mathbb{E}_{s' \sim p}[\tau \mathcal{H}(\pi(\cdot|s')) + \mathbb{E}_{a' \sim \pi(\cdot|s')}(Q^\pi(s', a'))]$$

  - 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:

$$Q(s_t, a_t) \leftarrow r(s_t, a_t) + \gamma \mathbb{E}_{s_{t+1}\sim\rho}[V(s_{t+1})], \forall s_t, a_t$$
$$V(s_t) \leftarrow \tau \log(\int_{\mathcal{A}} \exp(\frac{1}{\tau} Q(s_t, a'))\, da'), \forall s_t$$

  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 $\tilde{\pi}(\cdot|s) \propto \exp(Q^{\pi}_{soft}(s, \cdot))$ is intractable in general.

Algorithm: Soft Q-Iteration

- Proof:
  - We show that the soft value iteration operator $\mathcal{T}$, defined as:

$$\mathcal{T}Q(s,a) = r(s,a) + \gamma \mathbb{E}_{s'\sim p_s}[\log \int \exp Q(s',a')da']$$

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

$$\begin{aligned}\log \int \exp(Q_1(s',a'))da' &\le \log \int \exp(Q_2(s',a') + \varepsilon)da' \\ &= \log(\exp(\varepsilon)\int \exp Q_2(s',a')da') \\ &= \varepsilon + \log \int \exp(Q_2(s',a'))da'\end{aligned}$$

Therefore, $\|\mathcal{T}Q_1 - \mathcal{T}Q_2\| \le \gamma\varepsilon = \gamma\|Q_1 - Q_2\|$. So $\mathcal{T}$ 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: θ, ϕ

```
θ̄ ← θ, φ̄ ← φ
D ← ∅
for each epoch do
    for each t do
        a_t = f^φ(ξ, s_t), where ξ ~ N(0, I)
        s_{t+1} ~ p(s_{t+1} | s_t, a_t)
        D ← D ∪ {(s_t, a_t, r(s_t, a_t), s_{t+1})}
        Update soft Q-function parameters
        Update policy
    end for
    if epoch % update_interval = 0 then
        θ̄ ← θ, φ̄ ← φ
    end if
end for
return θ, φ
```

Tuomas Haarnoja et al. "Reinforcement Learning with Deep Energy-Based Policies". In:Proceedings of the 34th International Conference on Machine Learning, ICML 2017.

Algorithm: Soft Q-learning

- Firstly, we model the soft Q-function with a function approximator with parameters θ, $Q_\theta(s_t, a_t)$. We will optimize the soft Bellman error $|\mathcal{T}Q - Q|$ to find optimal Q-function.
- Update soft Q-learning parameters:
  1. Sample a minibatch from the replay memory: $\{s_t^{(i)}, a_t^{(i)}, r_t^{(i)}, s_{t+1}^{(i)}\}_{i=0}^{N} \sim \mathcal{D}$
  2. Sample M uniform actions: $\{a^{(i,j)}\}_{j=0}^{M} \sim q_{a'}$
  3. Compute empirical soft value with next equation:
  $$V_{\bar{\theta}}(s_{t+1}^{(i)}) = \tau \log \mathbb{E}_{q_{a'}}\left[\frac{\exp(\frac{1}{\tau}Q_\theta(s_t, a'))}{q_{a'}(a')}\right]$$
  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 Sort Q-iteration):
  $$g_\theta^{(i)} \leftarrow \nabla_\theta\left(\mathbb{E}_{s_t \sim q_s, a_t \sim q_a}\left[\frac{1}{2}(Q_\theta(s_t, a_t) - (r(s_t, a_t) + \gamma \mathbb{E}_{s_{t+1}\sim p}[V_{\bar{\theta}}(s_{t+1})]))^2\right]\right)$$
  5. Update θ: $\theta \leftarrow ADAM(\theta, g_\theta^{(i)})$

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:

$$\triangle f^\phi(\cdot; s_t) = \mathbb{E}_{a_t \sim \pi_\phi}[\kappa(a_t, f_\phi(\cdot; s_t))\nabla_{a'} Q_\theta(s_t, a')|_{a'=a_t} + \tau \nabla_{a'} \kappa(a', f^\phi(\cdot; s_t))|_{a'=a_t}]$$

  4. Compute empirical estimate $g_\phi^{(i)} = \hat{\nabla}_\phi J_\pi$ of next equation:

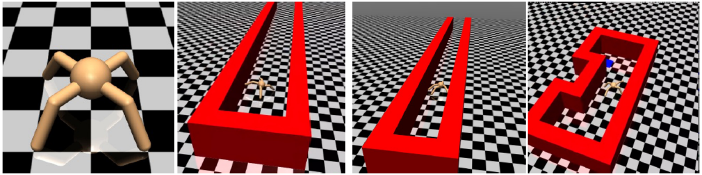
$$\frac{\partial J_\pi(\phi; s_t)}{\partial \phi} \propto \mathbb{E}_\xi \left[\triangle f^\phi(\xi; s_t) \frac{\partial f^\phi(\xi, s_t)}{\partial \phi} \right]$$

$$J_\pi(\phi; s_t) = D_{KL}\left(\pi_\phi(\cdot|s_t) \| \exp(\frac{1}{\tau}\left(Q_\theta(s_t, \cdot) - V_\theta(s_t)\right)\right)$$

  5. Update ϕ: $\phi \leftarrow ADAM(\phi, g_\phi^{(i)})$

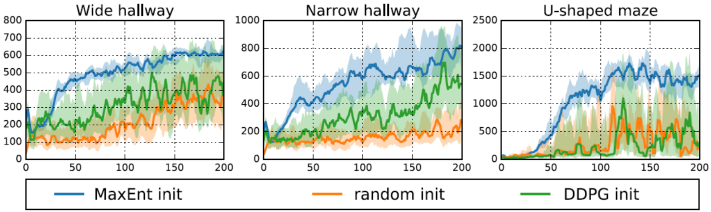
Algorithm: Soft Q-learning

- Experimental environment:



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

- Experimental environment:



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 $\mathcal{A}$:

$$V^*_{soft} = \mathcal{F}_\tau(r)$$

An important assumption: reward function of action, $r(a)$, are from σ^2-subgaussian.

Xiao, Chenjun, et al. "Maximum entropy monte-carlo planning." Advances in Neural Information Processing Systems. 2019.

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 π^* asymptotically. The algorithm selects an action by sampling from the distribution:

$$\pi_t(a) = (1-\lambda_t)\mathbf{f}_\tau(\mathbf{q})(a) + \lambda_t \frac{1}{|\mathcal{A}|}$$

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

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

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

where t is number of iterations, σ 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:

$$\pi_t(a|s) = (1-\lambda_s)\mathbf{f}_\tau(\mathbf{Q}_{soft}(s))(a) + \lambda_s \frac{1}{|\mathcal{A}|}$$

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

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

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:

$$\mathbb{P}(a_t \neq a^*) \leq Ct \exp\left(-\frac{t}{(\log t)^3}\right)$$

- Experimental effect:

Table 1: Performance comparison of Atari games playing.

Agent	*BeamRider*	*Breakout*	*Q*bert*	*Seaquest*	*SpaceInvaders*
DQN	19280	345	14558	1142	625
UCT	21952	367	16010	1129	656
MENTS	18576	**386**	**18336**	1161	**1503**

Other Algorithm I

1. Path Consistency Learning (PCL):
   Reference: Nachum, Ofir, et al. "Bridging the gap between value and policy based reinforcement learning." Advances in Neural Information Processing Systems. 2017.
2. Prove the equivalence of soft Q-learning and policy gradients
   Reference: Schulman, J., Abbeel, P., and Chen, X. Equivalence between policy gradients and soft Q-learning. arXiv preprint arXiv:1704.06440, 2017.
3. Hierarchical maximum entropy reinforcement learning
   Reference: Haarnoja, T., Hartikainen, K., Abbeel, P., and Levine, S. Latent space policies for hierarchical reinforcement learning. In International Conference on Machine Learning (ICML), 2018.
4. Combine Soft Q-learning policies to produce better policy
   Reference: Haarnoja, T., Pong, V., Zhou, A., Dalal, M., Abbeel, P., and Levine, S. Composable deep reinforcement learning for robotic manipulation. In International Conference on Robotics and Automation (ICRA). IEEE, 2018.

Other Algorithm II

5. Soft Actor-Critic algorithm
   Reference: Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In International Conference on Machine Learning (ICML), 2018.
6. Dynamically and automatically tuning the temperature parameter in Soft Actor-Critic Algorithm
   Reference: Haarnoja, Tuomas, et al. "Soft actor-critic algorithms and applications." arXiv preprint arXiv:1812.05905. 2018.
7. Soft policy gradient method
   Reference: Shi, Wenjie et al. "Soft Policy Gradient Method for Maximum Entropy Deep Reinforcement Learning." IJCAI (2019).

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

Reference

Note: Papers in this area mainly come from UC Berkeley, U alberta.

1 Nachum, Ofir, et al. "Bridging the gap between value and policy based reinforcement learning." Advances in Neural Information Processing Systems. 2017.

2 Schulman, J., Abbeel, P., and Chen, X. Equivalence between policy gradients and soft Q-learning. arXiv preprint arXiv:1704.06440, 2017.

3 Tuomas Haarnoja et al. "Reinforcement Learning with Deep Energy-Based Policies". In:Proceedings of the 34th International Conference on Machine Learning, ICML 2017.

4 Haarnoja, Tuomas. Acquiring Diverse Robot Skills via Maximum Entropy Deep Reinforcement Learning. Diss. UC Berkeley, 2018.

5 Xiao, Chenjun, et al. "Maximum entropy monte-carlo planning." Advances in Neural Information Processing Systems. 2019.

Thankyou!