A Handout on EXTREME Q-LEARNING

YU-CHENG CHEN
Department of Computer Science
National Yang Ming Chiao Tung University
yucheng.cs11@nycu.edu.tw

1 Introduction

Maximum Entropy (MaxEnt) policies are methods that are based on Regularized MDP. They are widely used in many popular RL algorithms, such as Soft Q-Learning (SQL)[1] and Soft Actor Critic (SAC)[2]. However, the Bellman backup used in MaxEnt RL algorithms requires computing the log-partition function over Q-values, which is usually intractable in high-dimensional action spaces. Specifically, we do V_{soft} and Q_{soft} backup:

$$Q_{soft}(s_t, a_t) \leftarrow r_t + \gamma E_{s_{t+1} \sim p_s}[V_{soft}(s_{t+1})], \forall s_t, a_t \tag{1}$$

$$V_{soft}(s_t) \leftarrow \alpha \log \int_A \exp(\frac{1}{\alpha} Q_{soft}(s_t, a'))da', \forall s_t \tag{2}$$

The researchers of SQL proofed that iteratively updating V_{soft} and Q_{soft} using above equations, Q and V converges to Q^*_{soft} and V^*_{soft} respectively. We noticed that integration over action in Eq.2 can't be calculated when the action space is continuous. Instead, current methods like SAC rely on auxiliary policy networks, as a result do not perform the optimal Bellman backup. In this paper, they directly model the soft-optimal values V^* by using the method called "Gumbel Regression". This new algorithm can ensure that V converges to the optimal solution V^*_{soft}, but it does not use the backup operation described above. In this paper, they outline the theoretical motivation for using Gumbel distributions in reinforcement learning, and show how it can be used to derive practical online and offline MaxEnt RL algorithms.

The main contributions of this paper are as follows:

- They motivate Gumbel Regression and show it allows calculation of the log-partition function (LogSumExp) in continuous spaces. They apply it to MDPs to present a novel loss objective for RL using maximum-likelihood estimation.
- Their formulation extends soft-Q learning to offline RL as well as continuous action spaces without the need of policy entropies. It allows us to compute optimal soft-values V^* and soft-Bellman updates using SGD, which are usually intractable in continuous settings.
- They empirically demonstrate strong results in Offline RL, improving over prior methods by a large margin on the D4RL Franka Kitchen tasks, and performing moderately better than SAC and TD3[3] in Online RL.

In my perspective, they interpret Regularized MDP from a new viewpoint and use a simple backup method for updates, which can be applied to both online and offline RL settings. In this paper, they claim that their approach achieves state-of-the-art performance. I believe their mathematical derivation and methodology are highly suitable for learning and reference.

2 Preliminaries

2.1 Generalized Version of Maximum Entropy RL

Consider a generalized version of Maximum Entropy RL that augments the standard reward objective with the KL-divergence between the policy and a reference distribution μ:

$$E_\pi[\sum_{t=0}^{\infty} \gamma^t (r(s_t, a_t) - \beta \log \frac{\pi(a_t|s_t)}{\mu(a_t|s_t)})] \tag{3}$$

Where β is the regularization strength. When μ is uniform, this becomes the standard MaxEnt objective used in online RL. In offline RL setting, we choose μ to be the behavior policy π_D that generated the fixed dataset D. Consequently, this objective enforces a conservative KL-constraint on the learned policy, keeping it close to the behavior policy. We can then solve the optimal policy $\pi^*(a|s)$ and the optimal soft-value $V^*(s)$:

$$V^*(s) = \beta \log \sum_a \mu(a|s) \exp(\frac{Q(s,a)}{\beta}) \tag{4}$$

$$\pi^*(a|s) = \mu(a|s) \exp(\frac{(Q^*(s,a) - V^*(s))}{\beta}) \tag{5}$$

Proof provided in Section 5. Note that from above equations, we can view online MaxEnt RL as keeping the learned policy close to a uniform distribution, which gives the policy more stochasticity.

2.2 Extreme Value Theorem (Fisher-Tippett Theorem)

Maximum of i.i.d. samples from exponentially tailed distributions will asymptotically converge to the Gumbel distribution $G(\mu, \beta)$, which has PDF $p(x) = e^{-(z+e^{-z})}$ where $z = \frac{x-\mu}{\beta}$ with location parameter μ and scale parameter β.

2.3 Gumbel-Max Trick

In machine learning, we often parameterize a discrete distribution in terms of an unconstrained vector of numbers. For some vector $\theta \in \mathbb{R}^k$, we do softmax transformation:

$$\pi_k = \frac{e^{\theta_k}}{\sum_{k'=1}^{K} e^{\theta_{k'}}} \tag{6}$$

If we don't want to explicitly construct our distribution using the softmax transform, it turns out that there exists another method for achieving the same effect: the Gumbel-max trick.

For a set of unnormalized probabilities θ_k, we can draw a sample from the corresponding categorical distribution as follows: for each θ_k we add a sample $G^{(k)}$ from the standard Gumbel distribution, and then select the index with the maximum sum. That is:

$$I = \arg\max_k (\theta_k + G^{(k)}) \sim Cat(\pi) \tag{7}$$

which has same effect as applying softmax transformation and draw.

These properties lead into the McFadden-Rust model[4][5] of MDPs as we state below.

2.4 McFadden-Rust Model

An MDP following the standard Bellman equations with stochasticity in the rewards due to unobserved state variables will satisfy the soft-Bellman equations over the observed state with actual rewards $\overline{r}(s, a)$, given two conditions:

1. Additive separability (AS): observed rewards have additive i.i.d. Gumbel noise, i.e. $r(s, a) = \overline{r}(s, a) + \epsilon(s, a)$, with actual rewards $\overline{r}(s, a)$ and i.i.d. noise $\epsilon(s, a) \sim G(0, \beta)$.
2. Conditional Independence (CI): the noise $\epsilon(s, a)$ in a given state-action pair is conditionally independent of that in any other state-action pair.

Moreover, the converse also holds: Any MDP satisfying the Bellman equations and following a softmax policy, necessarily has any i.i.d. noise in the rewards with AS + CI conditions be Gumbel distributed. This result enable the view of a soft-MDP as an MDP with hidden i.i.d. Gumbel noise in the rewards.

The aforementioned previous works inspired the author to use a Gumbel distribution-based approach to solve the optimization problem of soft MDP. This new MaxEnt RL framework, called XQL[6], can avoid the issue of high-dimensional action spaces in Eq.2 and does not require sampling or calculating entropy from the policy. This enables the method to be applied in both online and offline RL settings.

3 Supporting Lemmas and Theoretical Analysis

Most of the approximation-based RL algorithms are minimizing the prediction error of Q or V value. They use Mean Square Error (MSE) to update, which implicitly assume that the errors are Gaussian. However, in Regularized MDPs' setting, the researchers of this paper found that Gumbel distribution can better fit the prediction error. And by minimizing the Gumbel objective function, we can get exactly the optimal soft-values V^*, which correspond to the Log-Sum-Exp term provided in Eq.2 and Eq.4. They refer to this new method as "Gumbel Regression". The table below compares the normal Gaussian model with the new Gumbel model:

	Gaussian	Gumbel
Assumption	$X_i = X_{pred} + Gaussian(0, \sigma^2)$ $\implies Gaussian(0, \sigma^2) = X_i - X_{pred}$	$X_i = X_{pred} - Gumbel(0, \beta)$ $\implies Gumbel(0, \beta) = X_{pred} - X_i$
PDF	$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ $\implies f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}$	$f(x) = \frac{1}{\beta} e^{-(\frac{(x-\mu)}{\beta} + e^{-(\frac{x-\mu}{\beta})})}$ $\implies f(x) = \frac{1}{\beta} e^{-(\frac{x}{\beta} + e^{-\frac{x}{\beta}})}$
Likelihood	$L(\theta) = \prod_i e^{-\frac{(x_i - x_{pred})^2}{2\sigma^2}}$	$L(\theta) = \prod_i e^{-(\frac{x_{pred} - x_i}{\beta} + e^{-(\frac{x_{pred} - x_i}{\beta})})}$
Log-Likelihood	$LL(\theta) = \sum_i -\frac{(x_i - x_{pred})^2}{2\sigma^2}$	$LL(\theta) = \sum_i -(\frac{x_{pred} - x_i}{\beta} + e^{-(\frac{x_{pred} - x_i}{\beta})})$
Objective (minimize)	$NLL(\theta) = \frac{1}{2\sigma^2} \sum_i (x_i - x_{pred})^2$	$NLL(\theta) = \sum_i [e^{(\frac{x_i - x_{pred}}{\beta})} - (\frac{x_i - x_{pred}}{\beta})]$

From the left hand side of this table, we can conclude that using MSE loss on prediction error implies that errors are Gaussian. To solve the right hand side optimization problem, we first sample x_i from a dataset D, then rewrite the objective:

$$L(h) = E_{x_i \sim D}[e^{\frac{(x_i - h)}{\beta}} - \frac{(x_i - h)}{\beta} - 1] \tag{8}$$

Note that we add -1 in this equation because when $x_i = h$, the loss should be 0. By letting $\frac{\partial L}{\partial h} = 0$, we can get when:

$$h = \beta \log E_{x_i \sim D}[e^{\frac{x_i}{\beta}}] \tag{9}$$

L has minimum. Proof provided in Section 5. It means that we can directly models the Log-Sum-Exp over Q by using the following $ExtremeV$ loss function:

$$J(V) = E_{s,a \sim \mu}[e^{(Q(s,a) - V(s))/\beta}] - E_{s,a \sim \mu}[(Q(s,a) - V(s))/\beta] - 1 \tag{10}$$

where $Q(s,a) - V(s)$ is the prediction error of $V(s)$. In the online setting, μ is the old policy (sampled from the replay buffer), whereas in the offline setting, it is π_D (sampled from a fixed offline buffer). The loss function above does not involve any current policy π, so this method theoretically doesn't need current policy. In conclusion, we can update V by this loss function.

4 Discussions

I believe the most significant contribution of this paper is the invention of a simple and theoretically supported update method for solving Regularized MDP. Previous algorithms such as SQL and SAC had to bypass the high-dimensional action space problem using other methods, while the XQL algorithm introduced in this paper does not encounter this issue and can even update soft values

Algorithm 1 Extreme Q-learning ($\mathcal{X}$-QL) (Under Stochastic Dynamics)

1: Init Q_ϕ, V_θ, and π_ψ
2: Let $\mathcal{D} = \{(\mathbf{s}, \mathbf{a}, r, \mathbf{s}')\}$ be data from $\pi_\mathcal{D}$ (offline) or replay buffer (online)
3: **for** step t in {1...N} **do**
4: Train Q_ϕ using $\mathcal{L}(\phi)$ from Eq. 14
5: Train V_θ using $\mathcal{J}(\theta)$ from Eq. 11 (with $\mathbf{a} \sim \mathcal{D}$ (offline) or $\mathbf{a} \sim \pi_\psi$ (online))
6: Update π_ψ via Eq. 12 (offline) or Eq. 13 (online)
7: **end for**

Figure 1: Extreme Q-learning Algorithm

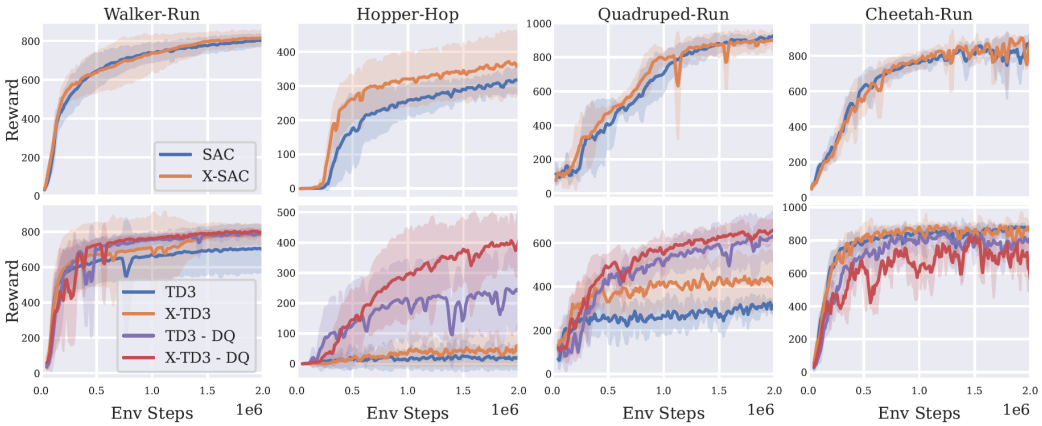

Figure 2: Results on the DM Control for SAC and TD3 based versions of Extreme Q Learning

without considering the policy. This not only simplifies the computation but also avoids many potential issues.

In practice, the authors designed a simple and highly efficient RL algorithm, which I have attached a screenshot (Fig.1) from the paper for your reference. Please refer to the original text for details.

Additionally, they combined this new update method with TD3 and SAC, resulting in "X-SAC" and "X-TD3," both of which achieved better performance than their original counterparts. I have also attached screenshots of the experimental results (Fig.2), and please refer to the original text for details on the experimental setup.

5 Proof

5.1 Proof of Generalized Version of Maximum Entropy RL

In regularized MDP, we want to maximize:

$$E_{a\sim\pi(.|s)}[Q(s,a)] + \beta H \tag{11}$$

where H is the entropy term. Let $H = -D_{KL}(\pi, \mu)$, where μ is any referenced distribution. Then rewrite the objective:

$$E_{a\sim\pi(.|s)}[Q(s,a)] - \beta D_{KL}(\pi, \mu) \tag{12}$$

Assume a is discrete, then we get:

$$\sum_a \pi(a|s)Q(s,a) - \beta \sum_a \pi(a|s) \log \frac{\pi(a|s)}{\mu(a|s)} \tag{13}$$

$$\sum_a \pi(a|s)[Q(s,a) - \beta \log \frac{\pi(a|s)}{\mu(a|s)}] \tag{14}$$

We can further write the above optimization problem:

$$maximize: \sum_a \pi(a|s)[Q(s,a) - \beta \log \frac{\pi(a|s)}{\mu(a|s)}], s.t. \sum_a \pi(a|s) = 1, \mu(a|s) \geq 0, \forall s, a \tag{15}$$

To solve this, we leverage the Lagrange multiplier technique:

$$L = \sum_a \pi(a|s)[Q(s,a) - \beta \log \frac{\pi(a|s)}{\mu(a|s)}] - m(\sum_a \pi(a|s) - 1) \tag{16}$$

where $m \in \mathbb{R}$ is the Lagrange multiplier. Then the optimal solution satisfies $\frac{\partial L}{\partial \pi(a|s)} = 0$ for every $a \in A$. L can be wrote as:

$$L = \begin{cases} \pi(a_1|s)Q(s,a_1) - \beta\pi(a_1|s) \log \frac{\pi(a_1|s)}{\mu(a_1|s)} - m\pi(a_1|s) \\ \pi(a_2|s)Q(s,a_2) - \beta\pi(a_2|s) \log \frac{\pi(a_2|s)}{\mu(a_2|s)} - m\pi(a_2|s) \\ \qquad\qquad ... \\ \pi(a_n|s)Q(s,a_n) - \beta\pi(a_n|s) \log \frac{\pi(a_n|s)}{\mu(a_n|s)} - m\pi(a_n|s) \end{cases} \tag{17}$$

We then can get system of equations:

$$\begin{cases} \frac{\partial L}{\partial \pi(a_1|s)} = Q(s,a_1) - \beta(\log \frac{\pi(a_1|s)}{\mu(a_1|s)} + 1) - m = 0 \\ \frac{\partial L}{\partial \pi(a_2|s)} = Q(s,a_2) - \beta(\log \frac{\pi(a_2|s)}{\mu(a_2|s)} + 1) - m = 0 \\ \qquad\qquad ... \\ \frac{\partial L}{\partial \pi(a_n|s)} = Q(s,a_n) - \beta(\log \frac{\pi(a_n|s)}{\mu(a_n|s)} + 1) - m = 0 \\ \pi(a_1|s) + \pi(a_2|s) + ... + \pi(a_n|s) = 1 \end{cases} \tag{18}$$

Solve these equations, we get:

$$\begin{cases} \pi(a_1|s) = \frac{\mu(a_1|s)\exp(Q(s,a_1)/\beta)}{\exp((m+\beta)/\beta)} \\ \pi(a_2|s) = \frac{\mu(a_2|s)\exp(Q(s,a_2)/\beta)}{\exp((m+\beta)/\beta)} \\ \qquad\qquad ... \\ \pi(a_n|s) = \frac{\mu(a_n|s)\exp(Q(s,a_n)/\beta)}{\exp((m+\beta)/\beta)} \end{cases} \tag{19}$$

By using $\pi(a_1|s) + \pi(a_2|s) + ... + \pi(a_n|s) = 1$, we get:

$$\frac{\mu(a_1|s)\exp(Q(s,a_1)/\beta)}{\exp((m+\beta)/\beta)} + \frac{\mu(a_2|s)\exp(Q(s,a_2)/\beta)}{\exp((m+\beta)/\beta)} + ... + \frac{\mu(a_n|s)\exp(Q(s,a_n)/\beta)}{\exp((m+\beta)/\beta)} = 1 \tag{20}$$

$$\mu(a_1|s)\exp(Q(s,a_1)/\beta) + \mu(a_2|s)\exp(Q(s,a_2)/\beta) + ... + \mu(a_n|s)\exp(Q(s,a_n)/\beta) = \exp((m+\beta)/\beta) \tag{21}$$

By replacing $\exp(\frac{m+\beta}{\beta})$ back to Eq.14, we get:

$$\begin{cases} \pi(a_1|s) = \frac{\mu(a_1|s)\exp(Q(s,a_1)/\beta)}{\mu(a_1|s)\exp(Q(s,a_1)/\beta) + \mu(a_2|s)\exp(Q(s,a_2)/\beta) + ... + \mu(a_n|s)\exp(Q(s,a_n)/\beta)} \\ \pi(a_2|s) = \frac{\mu(a_2|s)\exp(Q(s,a_2)/\beta)}{\mu(a_1|s)\exp(Q(s,a_1)/\beta) + \mu(a_2|s)\exp(Q(s,a_2)/\beta) + ... + \mu(a_n|s)\exp(Q(s,a_n)/\beta)} \\ \qquad\qquad ... \\ \pi(a_n|s) = \frac{\mu(a_n|s)\exp(Q(s,a_n)/\beta)}{\mu(a_1|s)\exp(Q(s,a_1)/\beta) + \mu(a_2|s)\exp(Q(s,a_2)/\beta) + ... + \mu(a_n|s)\exp(Q(s,a_n)/\beta)} \end{cases} \tag{22}$$

Then, the optimal π^* can be wrote as:

$$\pi^*(.|s) = \frac{\mu(.|s)\exp(Q^*(s,.)/\beta)}{\sum_a \mu(a|s)\exp(Q^*(s,a)/\beta)} \tag{23}$$

By defining $\exp(V^*(s)/\beta) = \sum_a \mu(a|s)\exp(Q^*(s,a)/\beta)$, we can get:

$$\pi^*(a|s) = \mu(a|s)\exp((Q^*(s,a) - V^*(s))/\beta) \tag{24}$$

and

$$V^*(s) = \beta \log \sum_a \mu(a|s)\exp(Q^*(s,a)/\beta) \tag{25}$$

5.2 Proof of Gumbel Regression Models the Log-Sum-Exp term

The objective of Gumbel Regression is:

$$L(h) = E_{x_i \sim D}[e^{\frac{(x_i - h)}{\beta}} - \frac{(x_i - h)}{\beta} - 1] \tag{26}$$

$\frac{\partial L}{\partial h} = 0$ gives:

$$E_{x_i \sim D}[e^{\frac{x_i - h}{\beta}}](-\frac{1}{\beta}) - E_{x_i \sim D}[-\frac{1}{\beta}] = 0 \tag{27}$$

$$\implies (-\frac{1}{\beta})e^{-\frac{h}{\beta}} E_{x_i \sim D}[e^{\frac{x_i}{\beta}}] = E_{x_i \sim D}[-\frac{1}{\beta}] = -\frac{1}{\beta} \tag{28}$$

Remove $-\frac{1}{\beta}$ both sides gives:

$$E_{x_i \sim D}[e^{\frac{x_i}{\beta}}] = e^{\frac{h}{\beta}} \tag{29}$$

Then we get:

$$h = \beta \log E_{x_i \sim D}[e^{\frac{x_i}{\beta}}] \tag{30}$$

References

[1] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1352–1361. PMLR, 06–11 Aug 2017.

[2] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1861–1870. PMLR, 10–15 Jul 2018.

[3] Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1587–1596. PMLR, 10–15 Jul 2018.

[4] Daniel McFadden et al. Conditional logit analysis of qualitative choice behavior. 1973.

[5] John Rust. Structural estimation of markov decision processes. *Handbook of econometrics*, 4:3081–3143, 1994.

[6] Divyansh Garg, Joey Hejna, Matthieu Geist, and Stefano Ermon. Extreme q-learning: Maxent rl without entropy. *arXiv preprint arXiv:2301.02328*, 2023.