Regularized Approaches for Soft MDP

Pu Yang
School of Mathematical Sciences, Peking University

2022.11.23

Table of contents

Markov Decision Processes (MDP) $\mathcal{M} =< \mathcal{S}, \mathcal{A}, P, R, \gamma >$[1]

- $\mathcal{S}$ - a set of states; $s \in \mathcal{S}$ - a state
- $\mathcal{A}$ - a set of actions; $a \in \mathcal{A}$ - an action (we only consider discrete case)
- P - transition probability function: $\mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathrm{R}$
- R - reward function: $\mathcal{S} \times \mathcal{A} \to \mathrm{R}$
- γ - discounting factor for future rewards

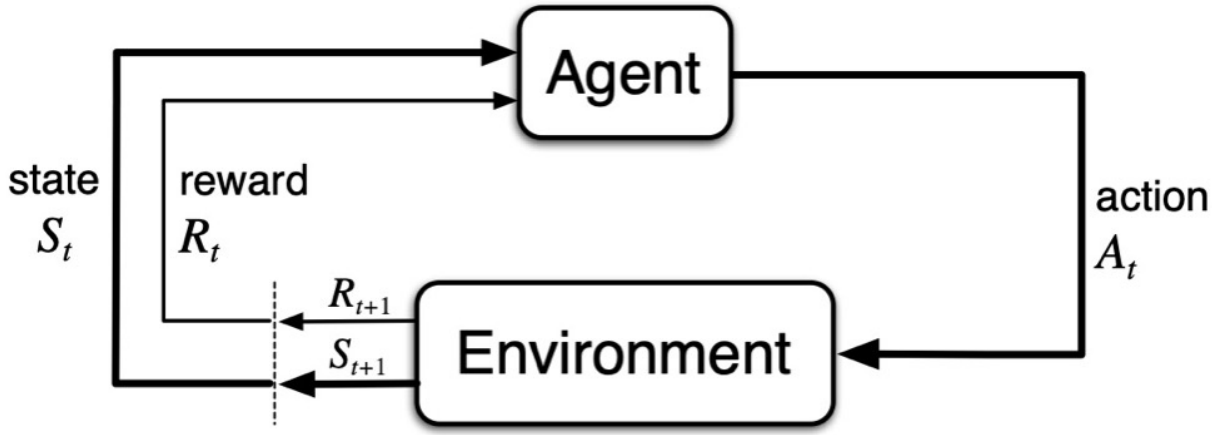


Figure 1: The agent–environment interaction in a Markov decision process.

[1] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Bellman Equation

- policy probability function: $\pi : \mathcal{S} \times \mathcal{A} \to \mathrm{R}$
- Value function: $V^{\pi}(s) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^k r_t \mid s_0 = s\right]$
- Q function: $Q^{\pi}(s,a) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^k r_t \mid s_0 = s, a_0 = a\right]$

Bellman equation

$$
\begin{aligned}
V^{\pi}(s) &= \mathbb{E}_{a\sim\pi}\left[Q(s_t,a) \mid s_t = s\right] \\
&= \mathbb{E}_{\pi}\left[r_t + \gamma V^{\pi}(s_{t+1}) \mid s_t = s\right]
\end{aligned}
$$

$$
\begin{aligned}
Q^{\pi}(s,a) &= \mathbb{E}\left[r_t + \gamma V^{\pi}(s_{t+1}) \mid s_t = s, a_t = a\right] \\
&= \mathbb{E}\left[r_t + \gamma \mathbb{E}_{a\sim\pi} Q^{\pi}(s_{t+1},a) \mid s_t = s, a_t = a\right]
\end{aligned}
$$

Bellman Optimality Equation

We aim to find the optimal policies by solving

$$\pi^* = \arg\max_{\pi} V^{\pi}(s)$$

- The optimal value function $V^* = V^{\pi^*}$ is unique solution of

  $$V^*(s) = \mathcal{T}V^*(s)$$

  where $\mathcal{T}$ is called Bellman Operator defined as

  $$\mathcal{T}V(s) = \max_{\pi} \mathbb{E}_{\pi}\left[r_{t+1} + \gamma V(s_{t+1}) \mid s_t = s\right]$$

- The optimal Q function $Q^* = Q^{\pi^*}$ satisfies

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

- The optimal policies (denoted as π^*) may not be unique

  $$\pi^*(a \mid s) = 0, \quad \text{if } Q(s,a) < \max_{a} Q(s,a)$$

  but π^* often is a deterministic policy which puts all probability mass on one action

Deterministic Policy vs. Stochastic Policy

In some cases, we might actually prefer to learn stochastic policies.
For example

- Multi-goal environment (see Fig. 2)
- Exploration (like DFS vs. BFS) (see Fig. 3)
- Robustness in the face of unstationary environment (may be future work)

Multi-goal Environment

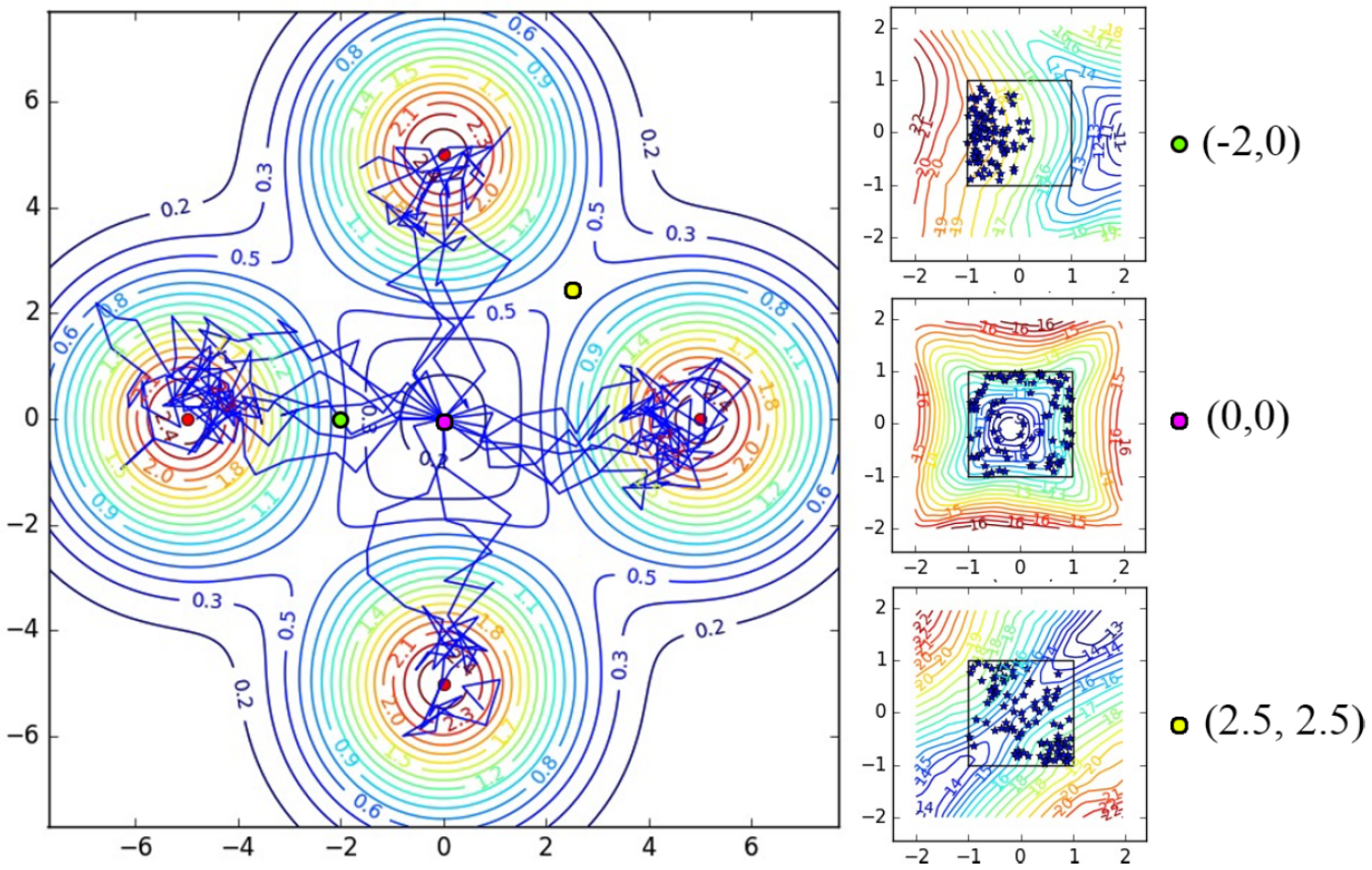


Figure 2: Illustration of the 2D multi-goal environment.

Exploration

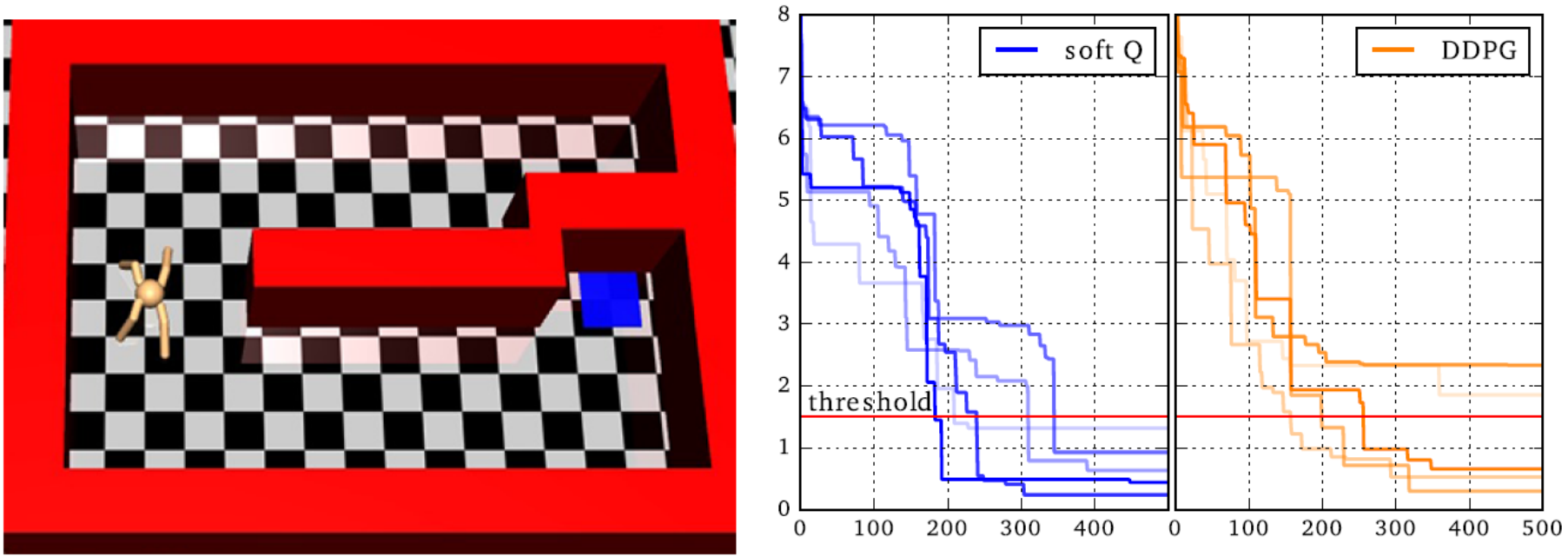


Figure 3: Comparison of soft Q-learning and DDPG the quadrupedal robot maze task.

Preliminary 1: Maximum Entropy Reinforcement Learning

- standard reinforcement learning objective

$$\pi^*_{\text{std}} = \arg\max_{\pi} \sum_t \mathbb{E}_{(s_t,a_t)\sim\rho_\pi} \left[r\left(s_t, a_t\right) \right]$$

- augments the reward with an entropy regularization term

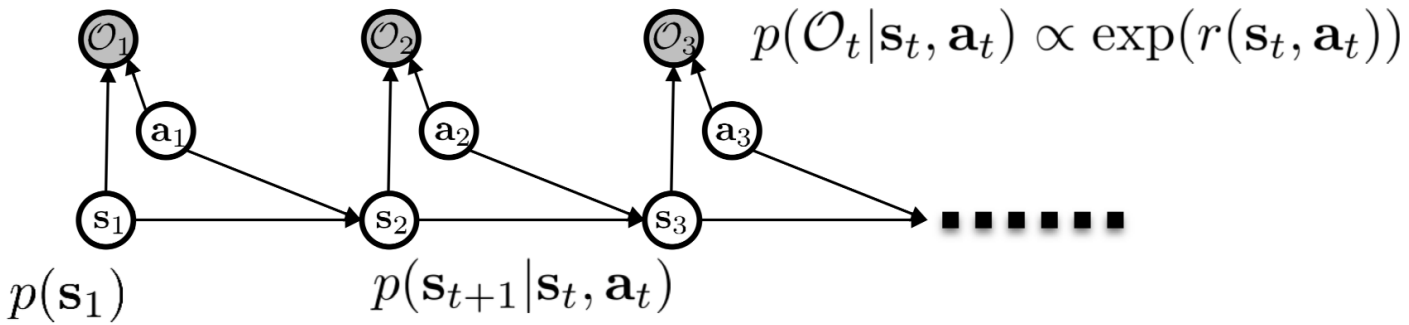
$$\pi^*_{\text{MaxEnt}} = \arg\max_{\pi} \sum_t \mathbb{E}_{(s_t,a_t)\sim\rho_\pi} \left[r\left(s_t, a_t\right) + \alpha \mathcal{H}\left(\pi\left(\cdot \mid s_t\right)\right) \right],$$

where α is the temperature parameter that controls the stochasticity of the policy, and the entropy is $\mathcal{H}(P) = \mathbb{E}_{x\sim P}[-\log P(x)]$

A Derivation from Probabilistic Graphical Model

We could consider control problems as the following temporal probabilistic graphical model (PGM)

$$p(\mathbf{s}_{1:T}, \mathbf{a}_{1:T}|\mathcal{O}_{1:T})$$



where $s_1, s_2, \ldots$ and $a_1, a_2, \ldots$ are hidden variables, representing states and actions, respectively; $O_1, O_2, \ldots$ are observed binary variables, which indicate whether the corresponding state and action are optimal.

A Derivation from Probabilistic Graphical Model

- Probabilistic inference problem: figure out the trajectory distribution given an optimal policy, i.e. $p(s_{1:T}, a_{1:T} \mid \mathcal{O}_{1:T})$
- We assume $O_{1:T} = \mathbf{1}$, which indicates the probability of a trajectory given that it is optimal

$$p(s_{1:T}, a_{1:T} \mid \mathcal{O}_{1:T}) \propto p(s_{1:T}, a_{1:T}) \exp\left(\sum_{t=1}^{T} r(s_t, a_t)\right)$$

A Derivation from Probabilistic Graphical Model

- Control via Variational Inference.
  Recall the ELBO

$$\log p\left(O_{1:T}\right) \geq \mathbb{E}_{s_{1:T}, a_{1:T} \sim q}\left[\log \frac{p\left(O_{1:T}, s_{1:T}, a_{1:T}\right)}{q\left(s_{1:T}, a_{1:T}\right)}\right]$$

  Because q is an arbitrary distribution, let

$$q\left(s_{1:T}, a_{1:T}\right) = p\left(s_1\right) \prod_{t=1}^{T} p\left(s_{t+1} \mid s_t, a_t\right) \pi\left(a_t \mid s_t\right)$$

  then

$$\log p\left(O_{1:T}\right) \geq \sum_{t=1}^{T} \mathbb{E}_{s_t, a_t \sim q}\left[r\left(s_t, a_t\right) + \mathcal{H}\left(\pi\left(a_t \mid s_t\right)\right)\right]$$

Preliminary 2: Soft Bellman Equation

- Soft Q function

$$Q^{\pi}_{\text{soft}}(s,a) = r(s,a) + \mathop{\mathbb{E}}_{s_t,a_t\sim\rho_\pi}\left[\sum_{t=1}^{\infty}\gamma^t\left(r\left(s_t,a_t\right) + \alpha H\left(\pi\left(\cdot \mid s_t\right)\right)\right) \mid s_0 = s, a_0 = a\right]$$

Soft Bellman equation

$$Q^{\pi}_{\text{soft}}(s,a) = r(s,a) + \gamma\mathbb{E}_\pi\left[Q^{\pi}_{\text{soft}}(s_{t+1},a_{t+1}) + \alpha\mathcal{H}(\pi(\cdot \mid s_{t+1})) \mid s_t = s, a_t = a\right]$$

- Energy Based Policy

Theorem

Given a policy π, define a new policy $\tilde{\pi}$ as

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

Assume that Q and $\int \exp(Q(s,a))da$ are bounded. Then

$$Q^{\tilde{\pi}(s,a)}_{soft} \geq Q^{\pi}_{soft}, \forall s, a$$

Soft Bellman Optimality Equation

The optimal soft Q function $Q^*_{\text{soft}}(s,a) = Q^{\pi^*_{\text{MaxEnt}}}_{\text{soft}}(s,a)$ satisfies

$$\pi^*_{\text{MaxEnt}} \propto \exp(\frac{1}{\alpha} Q^*_{\text{soft}})$$

Then we get the soft Bellman optimality equation

$$Q^*_{\text{soft}}(s_t, a_t) = r_t + \gamma \mathbb{E}_\pi [\alpha \cdot \text{softmax} \frac{1}{\alpha} Q^*_{\text{soft}}(s_{t+1}, \cdot)]$$

where

$$\text{softmax} Q(s,\cdot) = \log \int_{\mathcal{A}} \exp(Q(s,a)) da$$

Soft Q-learning (SQL)[2]

- We aim to minimize the soft Bellman error

$$\min_Q |\mathcal{T}Q - Q|$$

where

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

- We solve it via *soft policy iteration*

$$Q_{\text{soft}}^{(k+1)} \rightarrow \mathcal{T} Q_{\text{soft}}^{(k)}$$

- Theoretic guarantee: $Q_{\text{soft}}^{(k)}$ convergence to a local minima.

[2] Tuomas Haarnoja et al. "Reinforcement learning with deep energy-based policies". In: *International conference on machine learning*. PMLR. 2017, pp. 1352–1361.

Soft Policy Iteration in SQL

Similar to standard Q learning, we iteratively do the following three steps:

- Collect experience (s_t, a_t, s_{t+1}, r_t) from the environment with π
- Soft Policy Evaluation (update Q)

$$\hat{Q}(s_t, a_t) = r_t + \gamma \cdot \alpha \cdot \text{softmax} \frac{1}{\alpha} Q_{\text{soft}}(s_{t+1}, \cdot)$$

$$J_Q = \mathbb{E}_{(s_t, a_t, r_t, s_{t+1}) \sim \mathcal{D}} \frac{1}{2} (\hat{Q}(s_t, a_t) - Q(s_t, a_t))^2$$

  update Q with the gradient of J_Q

- Soft Policy Improvement (update π)

$$\pi(\cdot \mid s) \propto \exp(\frac{1}{\alpha} Q(s, \cdot))$$

Soft Bellman Equation

- Soft value function

$$V_{\text{soft}}^{\pi}(s) = \mathop{\mathbb{E}}_{s_t, a_t \sim \rho_\pi} \left[\sum_{t=0}^{\infty} \gamma^t \left(r\left(s_t, a_t\right) + \alpha H\left(\pi\left(\cdot \mid s_t\right)\right) \right) \mid s_0 = s \right]$$

Soft Bellman equation

$$\begin{aligned} Q_{\text{soft}}^{\pi}(s,a) &= r(s,a) + \gamma \mathbb{E}_\pi \left[(Q_{\text{soft}}^{\pi}(s_{t+1}, a_{t+1}) + \alpha \mathcal{H}(\pi(\cdot \mid s_{t+1}))) \mid s_t = s, a_t = a \right] \\ &= r(s,a) + \gamma \mathbb{E}_\pi \left[V_{\text{soft}}^{\pi}(s_{t+1}) \mid s_t = s, a_t = a \right] \end{aligned}$$

$$\begin{aligned} V_{\text{soft}}^{\pi}(s) &= \mathbb{E}_\pi \left[Q_{\text{soft}}^{\pi}(s_t, a_t) \mid s_t = s \right] + \alpha \mathcal{H}(\pi(\cdot \mid s)) \\ &= \mathbb{E}_\pi \left[Q_{\text{soft}}^{\pi}(s_t, a_t) - \alpha \log \pi(a_t \mid s_t) \mid s_t = s \right] \end{aligned}$$

Soft Actor-Critic (SAC)[3]

- We aim to minimize the soft Bellman error

$$\min_{Q} |\mathcal{T}Q - Q|$$

where

$$\mathcal{T}Q\left(s_t, a_t\right) \triangleq r\left(s_t, a_t\right) + \gamma \mathbb{E}_{s_{t+1} \sim p}\left[V\left(s_{t+1}\right)\right]$$

where

$$V\left(s_t\right) = \mathbb{E}_{a_t \sim \pi}\left[Q\left(s_t, a_t\right) - \log \pi\left(a_t \mid s_t\right)\right]$$

- We solve it via *soft policy iteration*

$$Q_{\text{soft}}^{(k+1)} \rightarrow \mathcal{T}Q_{\text{soft}}^{(k)}$$

- Theoretic guarantee: $Q_{\text{soft}}^{(k)}$ convergence to a local minima.

[3] Tuomas Haarnoja et al. "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor". In: *International conference on machine learning*. PMLR. 2018, pp. 1861–1870.

Soft Policy Iteration in SAC

- Collect experience
- Soft Policy Evaluation

$$\hat{Q}(s_t, a_t) = r_t + \gamma V(s_{t+1})$$
$$\hat{V}(s_t, a_t) = Q(s_t, a_t) - \alpha \log \pi(a_t \mid s_t)$$

$$J_Q = \frac{1}{2}\mathbb{E}_{(s_t, a_t, r_t, s_{t+1}) \sim \mathcal{D}}(\hat{Q}(s_t, a_t) - Q(s_t, a_t))$$
$$J_V = \frac{1}{2}\mathbb{E}_{(s_t, a_t, r_t, s_{t+1}) \sim \mathcal{D}}(\hat{V}(s_t, a_t) - V(s_t, a_t))$$

- Soft Policy Improvement

$$\pi(\cdot \mid s) \propto \exp(\frac{1}{\alpha} Q(s, \cdot))$$

Ablation: Stochastic vs. Deterministic Policy

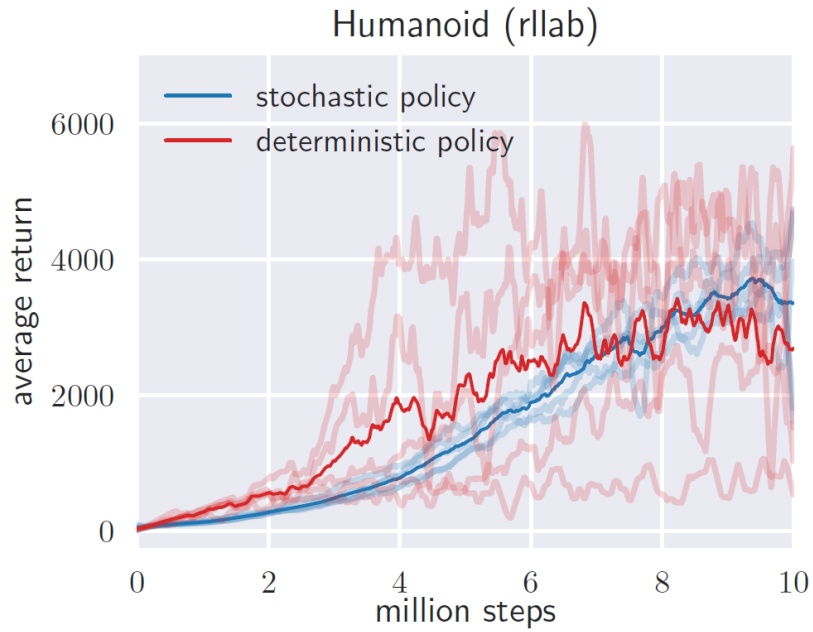
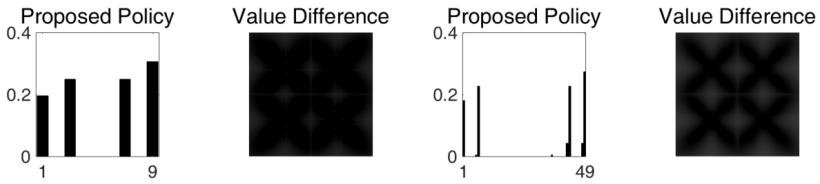


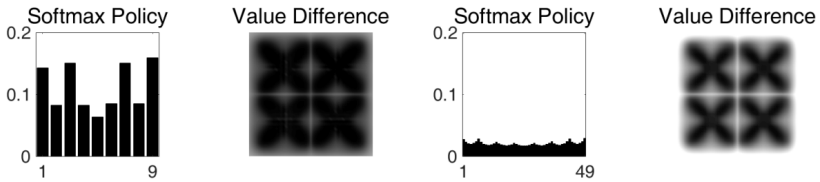
Figure 2. Comparison of SAC (blue) and a deterministic variant of SAC (red) in terms of the stability of individual random seeds on the Humanoid (rllab) benchmark. The comparison indicates that stochasticity can stabilize training as the variability between the seeds becomes much higher with a deterministic policy.

A Toy Example[4]

(a) Reward map and action values at state s.



(b) Proposed policy model and value differences (darker is better).



(c) Softmax policy model and value differences (darker is better).

Figure 4: A 2-dimensional multi-objective environment with point mass dynamics. The state is a location and the action is a velocity bounded with $[-3, 3] \times [-3, 3]$. The action space is discretized into two levels: 9 (low resolution) and 49 (high resolution).

From *Soft* to *Sparse*

- Soft: assigns a non-negligible probability mass to all actions
- Sparse: weeds out suboptimal actions and maintains near optimal actions

We want a sparse policy instead of a soft policy for two reasons:

- Safety: the policy gives bad actions with zero probability
- Performance

Regularized MDP[5]

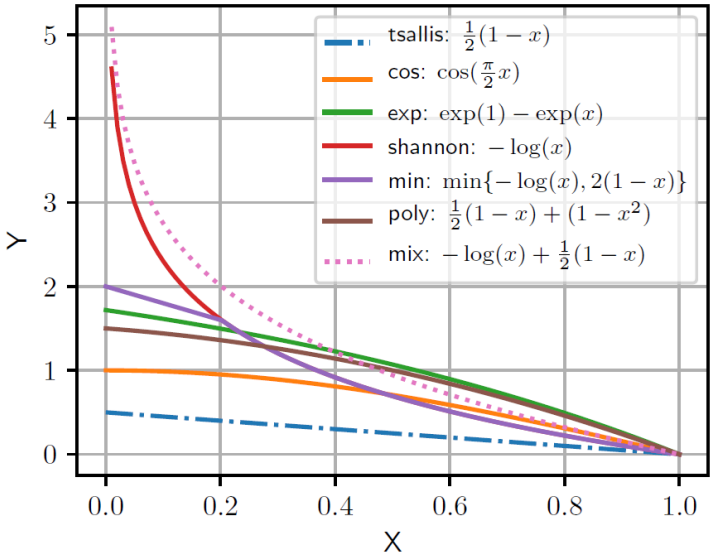
- Regularized MDP

$$\pi_\lambda^* = \arg\max_\pi \mathbb{E}_\pi \sum_{t=0}^{\infty} \gamma^t \left(r\left(s_t, a_t\right) + \lambda \phi\left(\pi\left(a_t \mid s_t\right)\right)\right)$$

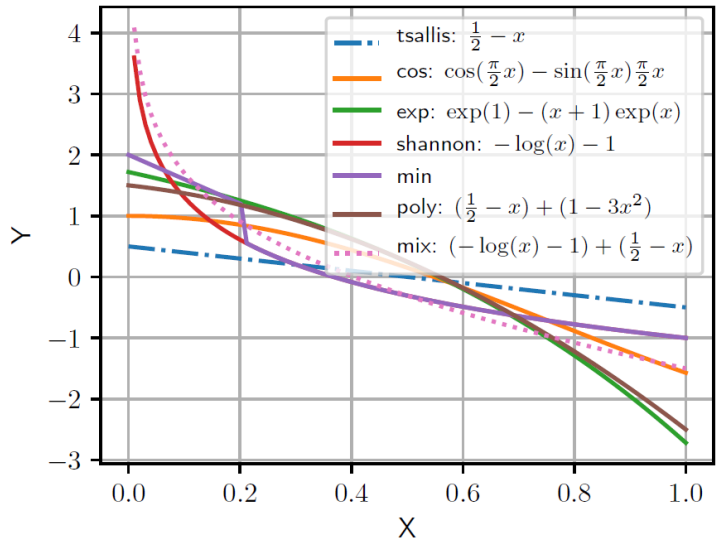
  where $\phi(x) \in [0, 1]$ is a regularization term.
- Assumption for the regularization term
  - Monotonicity: $\phi(x)$ is non-increasing
  - Non-negativity: $\phi(1) = 0$
  - Differentiability: $\phi(x)$ is differentiable;
  - **Expected Concavity**: $f_\phi(x) = x\phi(x)$ is stricly concave

[5] Wenhao Yang, Xiang Li, and Zhihua Zhang. "A regularized approach to sparse optimal policy in reinforcement learning". In: *Advances in Neural Information Processing Systems* 32 (2019).

Examples of Regularization Terms



(a) Different ϕ's



(b) f'_ϕ for different ϕ's

Bellman Optimality Equation

- Value function

$$V_\lambda^\pi(s) = \mathbb{E}_\pi \sum_{t=0}^{\infty} \gamma^t \left(r\left(s_t, a_t\right) + \lambda \phi\left(\pi\left(a_t \mid s_t\right)\right)\right)$$

- Q function and Bellman equation

$$\begin{aligned} Q_\lambda^\pi(s,a) &= r(s,a) + \gamma \mathbb{E} V_\lambda^\pi(s') \\ &= r(s,a) + \gamma \mathbb{E}_\pi [Q_\lambda^\pi(s',a') + \lambda \phi(\pi(a' \mid s'))] \end{aligned}$$

- Bellman Optimality Equation

$$\begin{aligned} Q_\lambda^*(s,a) &= r(s,a) + \gamma \mathbb{E} V_\lambda^*(s') \\ \pi^*(a \mid s) &= \max\left\{ g_\phi\left(\frac{\mu_\lambda^*(s) - Q_\lambda^*(s,a)}{\lambda}\right), 0 \right\} \\ V_\lambda^*(s) &= \mu_\lambda^* - \lambda \sum_a \pi_\lambda^*(a \mid s)^2 \phi'(\pi_\lambda^*(a \mid s)) \end{aligned}$$

where $g_\phi = (f'_\phi)^{-1}$

Definition and Property of Sparsity

Definition

A given policy π is called δ-sparse if it satisfies:

$$\frac{|\{(s,a) \in \mathcal{S} \times \mathcal{A} \mid \pi(a \mid s) \neq 0\}|}{|S||A|} \leq \delta$$

If $\pi(a \mid s) > 0$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}$, we call it has no sparsity.

Theorem

If $\lim_{x \to 0+} f'_\phi(x) = \infty$*, the optimal policy has no sparsity.*

Remark

- Proof is directly from the bellman equation
- Entropy regularization has no sparsity
- If $f'_\phi(0) < \infty$, we can control the sparsity with λ

Performance Error between V_λ^* and V^*

Under some assumptions, we have

$$\|V_\lambda^* - V^*\|_\infty \leq \frac{\lambda}{1-\gamma}\phi(\frac{1}{|\mathcal{A}|})$$

Regularized Actor-Critic (RAC)

- Collect experience
- Policy evaluation

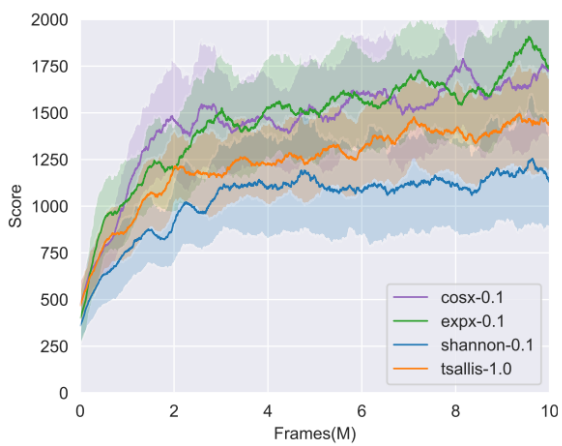
$$\hat{Q}(s_t, a_t) = r_t + \gamma(Q(s_{t+1}, a_{t+1}) + \lambda\phi(\pi(a_{t+1} \mid s_{t+1})))$$

$$J_Q = \frac{1}{2}\mathbb{E}_{\mathcal{D}}(\hat{Q} - Q)$$

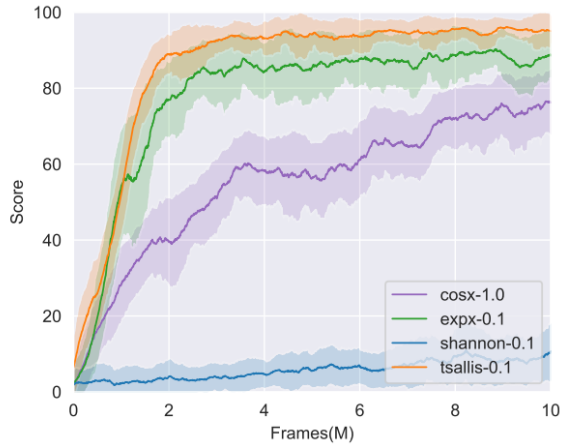
- Policy improvement

$$J_\pi = -\mathbb{E}_{\mathcal{D}}[Q + \lambda\phi(\pi)]$$

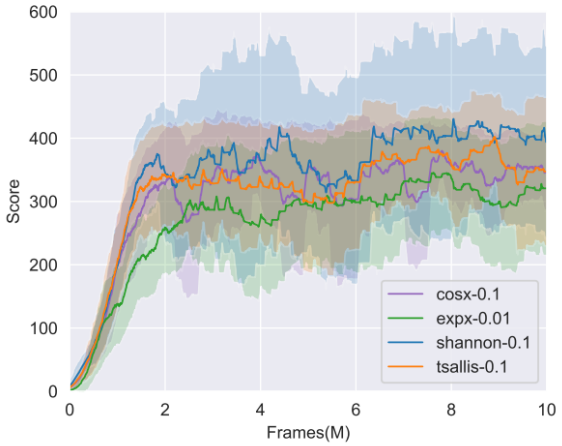
Experience on Atari games



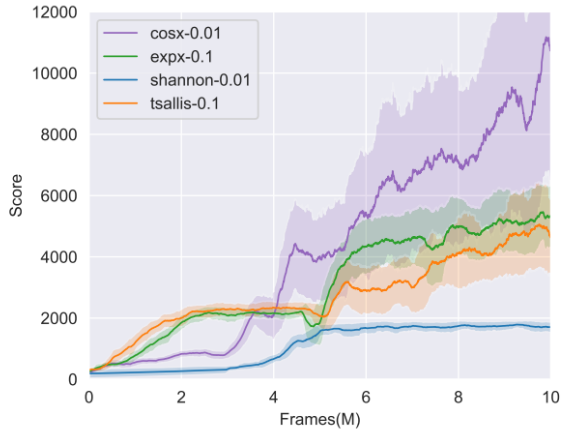
(a) Alien



(b) Boxing



(c) Breakout



(d) Seaquest

Background: mobile health (mHealth) technology

mHealth (mobile health) is a general term for the use of mobile phones and other wireless technology in medical care.

- Usages
  - educations about preventive healthcare services
  - disease surveillance
  - **treatment support**
  - epidemic outbreak tracking
  - **chronic disease management**
- Advantages: convenient
- Disadvantages
  - data breach
  - information may not be accurate
  - **safety**

Treatment Support[6]

We aim to monitor individuals' health statuses and deliver just-in-time personalized interventions. For example, monitor a diabetic's blood sugar, heart rate and other physiological indicators and decide the dose of insulin injection.

- large numbers of intervention options
- unclear what alternatives can be used (when meet temporary medication shortages)

[6] Wenzhuo Zhou, Ruoqing Zhu, and Annie Qu. "Estimating optimal infinite horizon dynamic treatment regimes via pt-learning". In: *Journal of the American Statistical Association* just-accepted (2022), pp. 1–40.

Methodology and Theory

- choose $\phi(x) = -\frac{1}{2}(x-1)$
- on-line $\rightarrow$ off-line, intuitively

$$\min_{\pi_\lambda} \|\mathcal{T}Q - Q\| \rightarrow \min_{\pi_\lambda} \sum_{i=1}^{n} \|\mathcal{T}Q(s^{(i)}, a^{(i)}) - Q(s^{(i)}, a^{(i)})\|$$

so we must consider the sample complexity

$$\|V_\lambda^{\hat{\pi}_\lambda} - V^*\|_{L^2}^2 = \mathcal{O}(n^{-1/2}) + \mathcal{O}(\lambda^2|\mathcal{A}|)$$

where $\hat{\pi}_\lambda$ is the empirical risk minimizer and n is the number of samples.
Comments of Statistics Professionals: This theoretic guarantee is very important in statistics, since we can not evaluate our policy in the real world.

Simulation

- State: blood glucose, Adiponectin, blood pressure
- Action: insulin injections (Yes/No), physical activity (No/Moderate/Strong) and dietary intake (Yes/No)
- Reward: linear combination of the three states
- Transition: a given model for generating data (unknown to the policy)

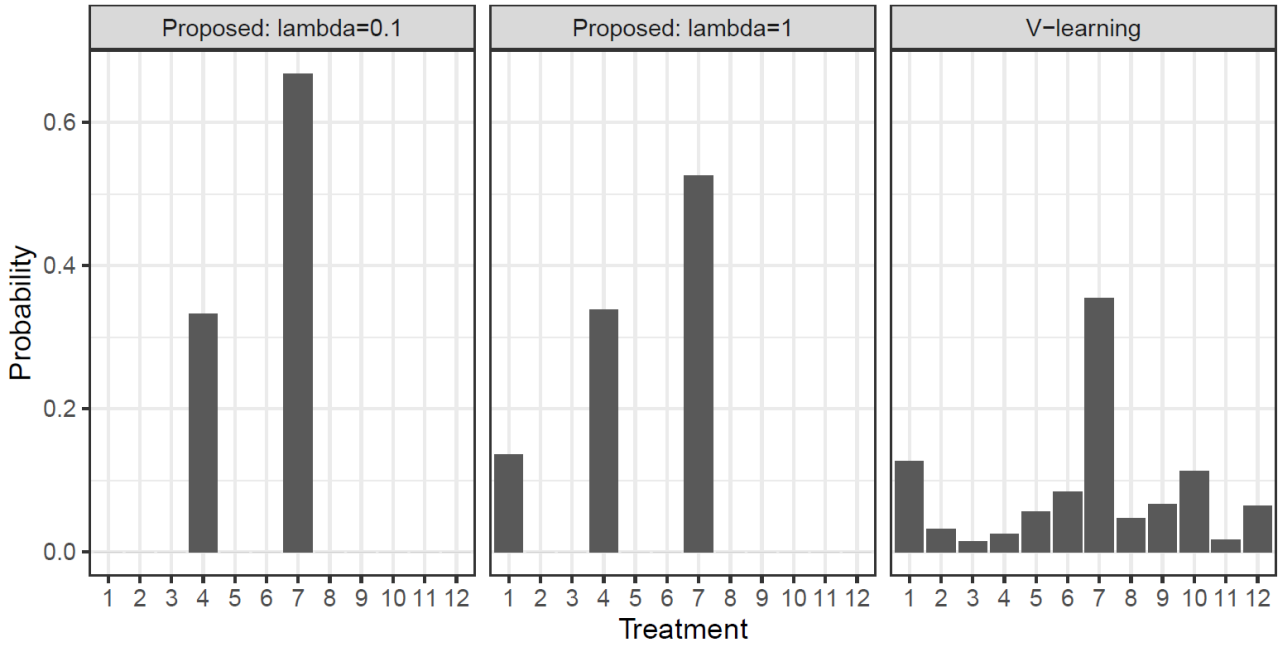


Figure 5: The estimated optimal policy distribution for one patient at a specific time point. The 7th treatment is the optimal treatment, and the 4th treatment is a near-optimal treatment; other treatments are sub-optimal or non-optimal.