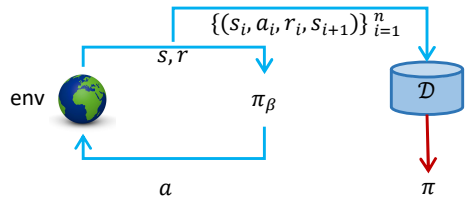
Learning to Trust Bellman Updates: Selective State-Adaptive Regularization for Offline RL

Qin-Wen Luo*, Ming-Kun Xie*, Ye-Wen Wang, and Sheng-Jun Huang
Nanjing University of Aeronautics and Astronautics, Nanjing, China

ICML International Conference On Machine Learning

Offline Reinforcement Learning

- Learn an effective policy π from a static dataset $\mathcal{D}$ collected by a behavior policy π_β



Motivation

To combat extrapolation errors, previous methods uniformly regularize the value function or policy updates across all data

- The optimal regularization intensity varies with the task, training process, and data density
- The improper global regularization hinders the efficiency of offline-to-online RL

Our method proposes state-adaptive regularization that dynamically quantifies the reliability of Bellman updates, guiding the policy to trust optimistic outcomes at the state level

- Enable the learned policy to benefit from the potential generalization of Bellman updates

The Proposed Method

Selective State-Adaptive Regularization (SSAR)

- State-Adaptive Coefficients

regularization coefficient $\beta \to \beta(s)$ neural-network-based state-wise coefficients

How to update $\alpha(s)$ for methods with different constraint objectives?

The regularization of CQL $\min_Q \beta\, \mathbb{E}_{s\sim D}\left[\log \sum_a \exp(Q(s,a)) - \mathbb{E}_{a\sim D}[Q(s,a)]\right]$

$$\Updownarrow\ \pi(a|s) \propto \exp(Q(s,a))$$

$$\min_\pi \beta\, \mathbb{E}_{(s,a)\sim D}[-\log \pi(a|s)]$$

The regularization directly affect the probabilities of dataset actions in the learned policy

A unified framework for updating coefficients in both value regularization and explicit policy constraint methods

$$L_\beta(\phi) = \mathbb{E}_{(s,a)\sim D}[\log \pi(a|s) - C_n(s)]\,\beta_\phi(s) \quad \textit{stochastic policy}$$
$$C_n(s) = \min\{\log \pi(\mu + n\sigma|s), \log \pi(\mu - n\sigma|s)\}$$

$$L_\beta(\phi) = \mathbb{E}_{(s,a)\sim D}[n^2\delta^2 - (a - \pi(s))^2]\beta_\phi(s) \quad \textit{deterministic policy}$$

- Distribution-Aware Thresholds

A simple linear schedule to dynamically adjust the threshold

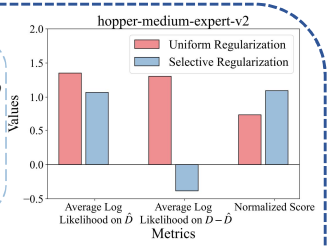
$$n =\leftarrow n + \Delta n$$
$$\Delta n = (n_{end} - n_{start}) \cdot T_{inc}/T$$

stop updating until $\mathbb{E}_{(s,a)\sim D}[\log \pi(a|s) - C_n(s)] > 0$

dynamically expand the trust region in a distribution-aware manner

- Selective Regularization

The proposed coefficient update mechanism encourages the policy to assign high probabilities to dataset actions, including suboptimal or poor ones, potentially resulting in degraded performance



sub-dataset selection for regularization and coefficient update

The sub-datasets should maximally cover the support of the offline dataset

- *datasets with low quality variances*

  select trajectories with returns greater than a selective threshold

- *datasets with a wide-ranging distribution*

Using the IQL paradigm to capture the relative value of the data

$$L_V = \mathbb{E}_{(s,a)\sim D}[L_2^\tau(Q(s,a) - V(s))]$$
$$L_Q = \mathbb{E}_{(s,a,s')\sim D}[(r(s,a) + \gamma V(s') - Q(s,a))^2]$$

select data with positive advantages to construct a sub-dataset

Prioritize good actions, relax on poor ones

- Efficient Offline-to-Online RL

$$\beta_{on}(s) = \min\{1 - \frac{N}{N_{end}}, 0\} \cdot \beta(s)$$

Given the generalization of the coefficient network, simple linear annealing enables efficient and stable online fine-tuning, while reducing reliance on retaining offline data

Experiments

We incorporated the proposed method with **CQL** and **TD3+BC** to conduct experiments

Offline performance comparison with backbone algorithms

Dataset	TD3+BC		CQL		Avg.	
	Base	Ours	Base	Ours	Base	Ours
halfcheetah-m-v2	48.3±0.2	**56.5±3.7**	47.1±0.2	**63.9±1.2**	47.7	**60.0**
hopper-m-v2	58.7±3.9	**101.6±0.4**	65.6±3.5	**89.1±9.7**	62.1	**95.4**
walker2d-m-v2	82.3±2.2	**87.9±2.4**	81.6±1.2	**84.9±1.7**	81.9	**86.4**
halfcheetah-mr-v2	44.4±0.6	**49.6±0.3**	45.7±0.4	**53.8±0.4**	45.0	**51.7**
hopper-mr-v2	66.4±27.1	**101.6±0.7**	92.3±9.3	**101.4±2.1**	79.3	**101.5**
walker2d-mr-v2	81.6±7.1	**93.5±2.0**	79.2±1.9	**94.7±3.3**	80.4	**94.1**
halfcheetah-me-v2	92.9±2.0	**94.9±1.2**	93.0±4.2	**102.1±1.2**	93.0	**98.5**
hopper-me-v2	101.4±8.2	**103.8±6.7**	97.8±8.6	**109.6±3.2**	99.6	**106.7**
walker2d-me-v2	110.3±0.5	**112.5±1.4**	109.2±0.2	**112.2±0.9**	109.8	**112.4**
halfcheetah-e-v2	**95.9±1.1**	95.5±1.3	97.0±0.5	**105.9±0.9**	96.5	**100.7**
hopper-e-v2	108.4±3.6	**109.8±4.3**	108.7±2.8	**111.4±0.2**	108.6	**110.6**
walker2d-e-v2	**110.1±0.5**	109.6±0.3	110.1±0.2	**110.2±0.2**	**110.1**	110.0
locomotion total	1000.8	**1116.7**	1030.4	**1139.1**	1015.6	**1128.0**
95% CIs	917.9∼1083.7	1096.2∼1137.3	990.4∼1070.1	1111∼1167.3	937.5∼1078.6	1093.2∼1162.8
umaze-v2	88.6±4.6	**93.4±3.3**	92.8±1.5	**96.0±2.3**	90.7	**94.7**
umaze-diverse-v2	43.2±18.8	**50.0±5.4**	27.8±13.1	**80.2±7.9**	35.5	**65.1**
medium-play-v2	0.0±0.0	**49.4±3.4**	67.0±4.2	**70.2±6.7**	33.5	**59.8**
medium-diverse-v2	0.0±0.0	**47.6±12.1**	60.5±9.2	**71.6±9.3**	30.3	**59.6**
large-play-v2	0.0±0.0	**18.0±4.6**	24.8±9.8	**53.0±4.1**	12.4	**35.5**
large-diverse-v2	0.0±0.0	**17.6±9.8**	21.2±12.1	**35.8±18.9**	10.6	**26.7**
antmaze total	131.8	**276.0**	294.1	**406.8**	213.0	**341.4**
95% CIs	78.2∼185.5	246.5∼305.5	230.9∼357.3	334.9∼478.7	130.1∼295.8	273.7∼409.1

Offline-to-online performance comparison with 250k interactions

Dataset	IQL	SPOT	FamCQL	CQL	TD3+BC	TD3+BC(SA)	CQL(SA)
halfcheetah-medium-v2	49.7	58.6	65.3	48.0	52.5	82.9±2.5	**95.3±1.5**
hopper-medium-v2	75.2	99.9	101.0	63.8	63.7	**103.5±0.4**	99.3±3.8
walker2d-medium-v2	80.8	82.5	93.3	82.8	86.6	101.6±7.4	**105.9±3.7**
halfcheetah-medium-replay-v2	45.2	57.6	73.1	49.4	49.3	73.1±3.0	**79.4±2.3**
hopper-medium-replay-v2	91.1	97.3	102.8	101.3	97.0	102.9±0.9	**103.1±0.2**
walker2d-medium-replay-v2	89.2	86.4	103.6	87.9	89.9	100.9±5.4	**116.3±2.1**
halfcheetah-medium-expert-v2	92.4	91.9	95.7	95.7	93.2	98.5±4.1	**115.4±1.5**
hopper-medium-expert-v2	109.6	106.5	104.4	110.8	99.8	**111.2±2.9**	109.5±5.4
walker2d-medium-expert-v2	115.0	110.6	110.4	109.8	115.8	115.7±5.5	**117.5±2.5**
halfcheetah-expert-v2	96.4	94.1	106.5	97.3	95.8	102.5±0.9	**113.3±0.8**
hopper-expert-v2	100.3	111.8	109.6	111.9	109.5	**112.0±2.4**	110.8±1.6
walker2d-expert-v2	112.5	109.9	112.6	109.7	111.4	**113.8±0.5**	112.6±1.2
locomotion total	1057.4	1107.1	1178.3	1068.4	1064.5	1218.6	**1278.4**
95% CIs min	981.5	1093.1	1165.3	1058.9	1039.8	1165.4	1254.9
95% CIs max	1133.2	1121.4	1191.5	1080.1	1089.2	1248.9	1303.5
antmaze-umaze-v2	83.0	98.8	-	95.2	72.8	96.5±3.2	**99.0±0.6**
antmaze-umaze-diverse-v2	38.2	56.8	-	59.2	39.8	87.2±5.0	**95.0±2.5**
antmaze-medium-play-v2	78.8	**92.5**	-	77.0	0.0	76.5±16.3	88.0±2.4
antmaze-medium-diverse-v2	80.2	87.0	-	84.0	0.2	63.0±36.1	**89.0±3.2**
antmaze-large-play-v2	42.8	60.0	-	51.8	0.0	35.5±13.2	**66.5±13.4**
antmaze-large-diverse-v2	40.2	**63.0**	-	38.2	0.0	30.5±15.0	56.8±18.2
antmaze total	363.2	458.1	-	405.5	112.8	389.2	**494.3**
95% CIs min	302.8	384.0	-	327.9	79.5	316.1	415.6
95% CIs max	423.7	534.0	-	483.1	146.1	462.4	573.4

Our method offers significant improvements over various baselines in both offline and offline-to-online settings