The Effects of Regularization on Robotic Performance

Raaid Elmelegy[1,2], Tianhao Wu[3], Jiantao Jiao[3]

[1]Transfer-to-Excellence Program, University of California, Berkeley
[2]Santa Monica College
[3]Department of Electrical Engineering and Computer Sciences, UC Berkeley

Abstract

- Existing offline Reinforcement Learning (RL) algorithms aim to leverage previously collected datasets to learn effective policies without further exploration. However, in practice, the distributional shift between the learned policy and the policy used to collect the data often leads to overestimation.
- A recently proposed offline RL algorithm (CQL), showed promising results by adding a regularization term to prevent overestimation.
- In our study we experimented with different regularization methods to improve on the implementation of the CQL algorithm.

Introduction

- In recent studies, the combination of RL and deep neural networks has produced promising results in automating a wide range of decision-making control tasks[1], [2].
- Real-world implementations of these methods still face two major challenges: the high sample complexity of on-policy RL algorithms and the brittle convergence properties of off-policy RL algorithms [3].
- Off-policy RL algorithms aim to make use of previously collected data without the need for further exploration, making them more suitable for certain tasks that otherwise would be expensive (e.g., in robotics, educational agents, or healthcare) or dangerous (e.g., in autonomous driving, or healthcare) [4].
- Recently proposed algorithms, such as SAC [5] and CQL [3], have demonstrated empirical success in achieving pessimism, leading to better sample efficiency and preventing overestimation.
- We believe that by experimenting with other regularization methods, we could achieve better results that are both computationally efficient and theoretically guaranteed to achieve pessimism.

Methods and Materials

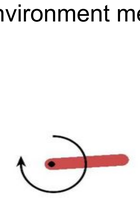
We applied regularization to the CQL algorithm's loss function to both achieve pessimism and avoid over-fitting.

$$L(\theta_i) = \alpha\mathbb{E}_{s_t\sim D}[\log \sum_a \exp Q_{\theta_i}(s_t, a) - \mathbb{E}_{a\sim D}[Q_{\theta_i}(s, a)] - \tau] + L_{SAC}(\theta_i)$$

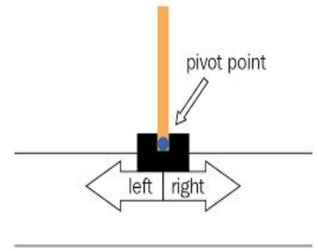
- First Method: Using the square of the Q function in the logarithmic exponential function.
- Second Method: Using the Square root of the Q function in the logarithmic exponential function.
- Third Method: Changing the loss function to a mean square function.

Results

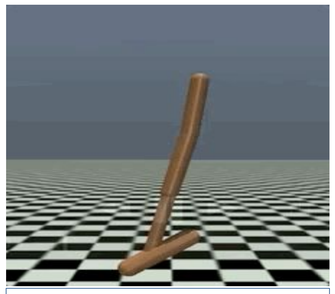
- We evaluated the algorithm on a variety of robotic tasks from the D4RL library.
- Methods 1 and 3 marginally outperformed the original implementation of CQL on the TD Error evaluation for the Hopper task.
- Method 2 outperformed CQL by a significant margin on the TD error evaluation and slightly underperformed on the environment metric.



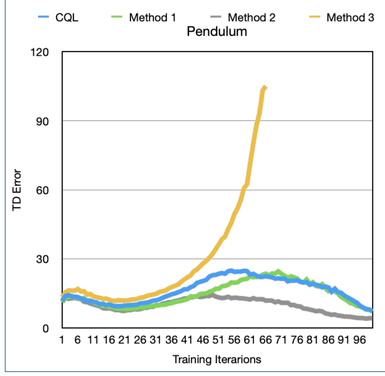
Pendulum

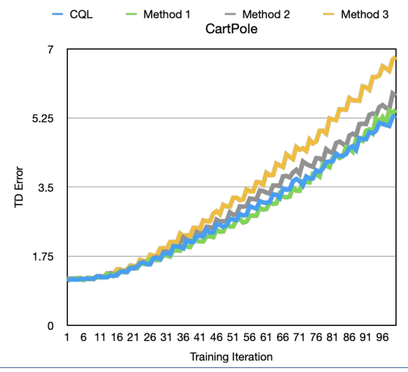


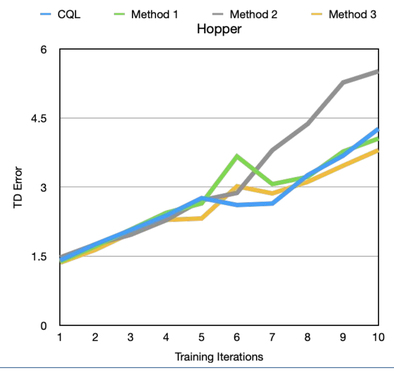
CartPole



Hopper







Discussion

- The most significant results were achieved in continuous domain tasks, particularly on the TD error metric.
- Although the results of our experiment are promising, we only able to evaluate our algorithm in a few D4RL environments and for a limited number of training iterations.

Conclusions and Future Directions

- The video rendering of our experiments suggest some improvement in performance in some cases but further testing and analysis is needed for a definitive conclusion.
- Testing our regularization methods on more tasks, including the D4RL library and other robotic task sumalators.
- Experimenting with other regularization methods.
- Investigating the reason behind our methods poor performance in discrete domain tasks.

Acknowledgements

- My mentor Tianhao Wu, for guiding me through the project and choosing this topic that I really enjoyed.
- Our PI Jiantao Jiao, for providing us with all the resources needed for the project.
- Transfer-to-Excellence program, for giving me the opportunity and the resources to work on this project.
- The Hopper-Dean Foundation, for funding this project.

Contact Information

References

[1] S. Gu, E. Holly, T. Lillicrap, and S. Levine, "Deep Reinforcement Learning for Robotic Manipulation with Asynchronous Off-Policy Updates." arXiv, Nov. 23, 2016. Accessed: Jul. 25, 2022. [Online]. Available: http://arxiv.org/abs/1610.00633
[2] T. Haarnoja, V. Pong, A. Zhou, M. Dalal, P. Abbeel, and S. Levine, "Composable Deep Reinforcement Learning for Robotic Manipulation." arXiv, Mar. 18, 2018. Accessed: Jul. 25, 2022. [Online]. Available: http://arxiv.org/abs/1803.06773
[3] A. Kumar, A. Zhou, G. Tucker, and S. Levine, "Conservative Q-Learning for Offline Reinforcement Learning," p. 13, [Online]. Available: https://doi.org/10.48550/arXiv.2006.04779
[4] S. Levine, A. Kumar, G. Tucker, and J. Fu, "Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems." arXiv, Nov. 01, 2020. Accessed: Jun. 14, 2022. [Online]. Available: http://arxiv.org/abs/2005.01643
[5] T. Haarnoja et al., "Soft Actor-Critic Algorithms and Applications." arXiv, Jan. 29, 2019. Accessed: Jul. 10, 2022. [Online]. Available: http://arxiv.org/abs/1812.05905
[6] J. Fu, A. Kumar, O. Nachum, G. Tucker, and S. Levine, "D4RL: Datasets for Deep Data-Driven Reinforcement Learning." arXiv, Feb. 05, 2021. Accessed: Aug. 07, 2022. [Online]. Available: http://arxiv.org/abs/2004.07219