# Conservative Q-Learning for Offline Reinforcement Learning

S Tarun Prasad (ME17B114) and Chella Thiyagarajan (ME17B179)

**Summary and main claims**: This paper adds value to reinforcement learning via novel theory. Authors of "Conservative Q-Learning for offline Reinforcement learning" paper have developed a new offline RL algorithm called conservative-Q-Learning(CQL). Authors claim that this algorithm performs very well compared to other offline RL algorithms out there. Naive Q learning propagates highly optimistic Q values while minimizing the bellman error, and the max reward operator in the bellman error seeks these errors out. CQL solves this issue with a single addition to the objective function. Loss of standard Q learning is:

$$L = E_{s,a\sim D}\left[\delta(s,a)\right] = E_{s,a\sim D}\left[||Q(s,a) - (r(s,a) + \gamma \max_{a'} Q(s',a'))||^2\right], \tag{1}$$

The paper suggests the following addition: $L_{cql} = E_{s,a\sim D}\left[\delta(s,a)\right] + \alpha \cdot E_{s\sim D,a\sim\pi}\left[Q(s,a)\right]$. This term forces all optimistic errors and forces Q values to be not greater than what they are supposed to be. The authors prove that with a proper choice of alpha, the resulting Q-function can be bounded by the "true" Q-values, and is thus a conservative estimate of those values. As a result our Q values now become too conservative empirically and the paper propose another addition to the loss function:

$$L_{cql} = E_{s,a\sim D}\left[\delta(s,a)\right] + \alpha \cdot (E_{s\sim D,a\sim\pi}\left[Q(s,a)\right] - E_{s,a\sim\pi}\left[Q(s,a)\right]) \tag{2}$$

Equation 2 tries to maximize the Q values of state actions that appear in the offline dataset, this encourages the agent to stick to more familiar actions and makes Q values less conservative. Our authors test this method on a wide range of RL benchmarks and show that this method is game changing in most ways.

**Merits of the Paper**: The proposed method is simple, tackles the right problem and operates in a very intuitive way. This algorithm can be easily incorporated on top of deep-Q-learning and actor critic algorithms with a very little overhead, it seems to be a plug and play algorithm. It has the advantage of not requiring a model for the data gathering policy, which eliminates a potential source of errors and removes redundant machinery and models from the process. The primary challenge in offline RL is successfully handling distributional shifts: learning effective skills requires deviating from the behavior in the data set and making counterfactual predictions about unseen outcomes. From the benchmarks we can see that the proposed algorithm seems to handle it very well compared to contemporary offline RL algorithms.

**Remarks and overall evaluation**: The paper focuses on constructing lower bounds but there is no concrete explanation about how conservative Q values are anywhere near actual true Q values. It is unclear what $\mu(a|s)$ denotes. In the preliminary section the author states that "Q-function training in offline RL does not suffer from state distribution shift" seems to be a wrong claim irrespective of it's explanation. The selection of alpha is not defined well. If the learned policy is drastically different from the behavioural policy then during test time the behaviour distribution can be far from test distribution over states and this again induces the problem of distribution shift. While standard Q-learning (and actor-critic) methods bootstrap from previous estimates, CQL is unique in that it is fundamentally a pessimistic algorithm, it assumes that if a good outcome was not seen for a given action, that action is likely to not be a good one.

**Clarity**: Preliminary section is explained well and introduces all the necessary concepts neatly and concisely. Explanations of some of the theorems are not explained well and it lacks many supporting claims which can be found in appendix. Overall the paper is very well explained except the theoretical proofs.

**Reproducibility and Real Time application**: Official implementation is made available in github public repository for all, therefore reproducibility is a sure thing. During Real time application, Fine-tuning a CQL agent is a non-trivial task, due to the so-called distribution shift problem – the agent encounters out-of distribution samples, and in turn loses its good initial policy from offline RL training.

Note: Theoretical proofs are really hard to understand, but it's easy to get a very good relevant idea about the problem it addresses and it's algorithmic solution.