A Comparative Study of Data-driven Offline Reinforcement Learning for Fed-batch Process Control

Omid Sobhania,\*, Furkan Elmaza, Michiel Robeynb, Johan Van den Hauwec, Shahab Pormoradi Gerdposhtehc, Benedict Cariusc, Kevin Metsa, Siegfried Mercelisa

aUniversity of Antwerp – imec, IDLab - Faculty of Applied Engineering, Sint-Pietersvliet 7, 2000 Antwerp, Belgium

bUniversity of Antwerp – imec, IDLab - Department of Computer Science, Sint-Pietersvliet 7, 2000 Antwerp, Belgium

callnex R&D, allnex, Anderlechtstraat 33, 1620 Drogenbos, Belgium

\* seyedomid.sobhani@uantwerpen.be

Abstract

Reinforcement Learning has gained traction in optimizing industrial processes; however, its application is hindered by safety concerns and the challenge of accurately simulating complex real-world scenarios. In response, Offline RL emerges as a promising approach, enabling agents to learn from historical data. This study investigates the use of Offline RL in a fed-batch chemical process, where the reward function is customized to train the agent to replicate optimal batches based on a reward function. Four state-of-the-art RL algorithms, including Conservative Q-Learning (CQL), and Batch Constrained Q-Learning (BCQ), Implicit Q-Learning (IQL), and Behavior Cloning (BC) are compared. This research examines the impact of dataset quality on RL agent performance, highlighting the importance of designing data collection experiments. Our results show that CQL outperforms other algorithms. Combining different recipe datasets reduces agent performance, while focusing solely on golden-batch experiments improves it at the expense of generalization. Overall, this study demonstrates the potential of Offline RL in controlling chemical processes.

**Keywords**: Offline-Reinforcement Learning, Data-driven control, Fed-batch process

* 1. Introduction

Reinforcement learning (RL) is a machine learning technique to learn an optimal policy to solve a sequential decision-making problem based on a reward function. The use of RL has received significant attention due to its impressive performance. This attention has spurred an increasing interest in implementing RL-based control and optimization algorithms in chemical processes (Ma, et al. 2019, Elmaz, et al. 2023). RL algorithms learn optimal policies by engaging in numerous interactions between agents and their environment (Sutton and Barto 2018). However, such interactions might not always be practical in industrial settings due to safety concerns and the lengthy nature of these processes. On the other hand, training within a simulation often suffers from the discrepancies between real-world scenarios and the simulation, due to the complexity of the process and potential inaccuracies. This can result in total failures when implementing the trained agent on-site.

To address these challenges, Offline RL has been introduced, allowing the agent to determine the best strategy based solely on logged data. While utilizing logged data helps minimize the simulation-to-reality (sim-to-real) gap, risks such as overfitting or bias in the dataset persist.

Additionally, such data may lack sufficient exploration, indicating the importance of data quality. In this study, we explore the utilization of offline RL techniques in a fed-batch industrial process scenario. The main objective of the designed agent is to control the process to follow predefined set-points by the human expert. We investigate the applicability of four state-of-the-art RL algorithms for optimizing the control policy of the process. Moreover, the study highlights how data quality impacts the performance of the trained agent. The remainder of this article is organized as follows: the experimental setup and data acquisition process is presented in section 2. The offline RL methodology is discussed in section 3. The results and conclusion are presented in section 4 and 5, respectively.

* 1. Reaction System and Data Collection

The process of interest includes a semi-batch (fed-batch) operating mode with a risk of uncontrolled side reactions causing failure. It's an exothermic reaction, demanding precise temperature control. The bath temperature and feed rate of one of the reagents are manipulated variables, while the inside temperature and the residual molar equivalents of the main reactants are the main controlled variables. Reactant molar equivalents are estimated via online near infra-red (NIR) spectroscopy. The impurities in the reagents and the reaction conditions, including the total feed length and mixture temperature affect the product quality. It is important to prevent the accumulation of side reactions during the reaction, as they can lead to the formation of a highly viscous product, which is undesired. The process involves a feeding phase and a post-feeding phase.

In this study, 44 lab-scale experiments were conducted with two recipes (recipe 1 and recipe 2) under different initial conditions and strategies, and measurements were taken each minute. The feeding time is the main difference between these two recipes. Among these, 4 are 'golden batch' experiments, optimized by human experts. Additionally, a simulation environment based on mass and heat balance equations was developed. Using this, and incorporating the control actions from the experiments, 44 simulated datasets were created, with the main difference being that observations are outputs from the simulation model. The agent is trained offline using experimental control strategies but observes outputs from the simulation. Validation occurs within the simulation environment, aiming to minimize observation discrepancies between training data and the validation environment.

It is important to note that due to the confidentiality of the reaction system, all chemical names have been masked, and the presented variables and graphs are normalized.

* 1. Offline Reinforcement Learning

Offline RL is a method where an agent learns exclusively from a fixed dataset without interacting with the environment (Levine, et al. 2020). The dataset comprises a set of transitions previously gathered, and no new data is collected during training. This can lead to the agent encountering states during deployment that it hasn't seen in training, causing extrapolation errors. This can cause the policy to overestimate the value of out-of-distribution states, leading to sub optimal actions with low rewards. While distributional shift can also occur in off-policy RL, the agent can mitigate and correct those over-estimations through environment interaction and learning from real-time experiences.

Conservative Q-Learning (CQL) (Kumar, Zhou, et al. 2020), Batch Constrained Q-Learning (BCQ) (Fujimoto, Meger and Precup, 2019), Implicit Q-Learning (IQL) (Kostrikov, Nair and Levine 2021), and Behavior Cloning (BC) (Fujimoto and Gu. 2021) are offline algorithms designed for static datasets, each with its approach to optimize policy learning. CQL and BCQ focus on constraining their learning to avoid overestimation and overfitting, with CQL being more conservative and BCQ focusing on actions’ distribution. IQL introduces robustness by considering a distribution of Q-values. In contrast, BC takes a more direct approach by simply replicating expert behaviors without long-term vision. For training the agent, the data is structured into a Markov Decision Process (MDP) with states including time step, component concentration, and reactor temperature, alongside their desired future values. The action space includes two continuous variables to manipulate the flow rate of component A and set point temperature. The reward at time step is defined as follows:

,

where and are weight factors to emphasize the importance of the elements in the reward functions. To investigate the performance of the trained agent, we used the simulation environment described in the previous section. The simulation model is wrapped as OpenAI Gym (Brockman, et al. 2016) environment and python library d3rlpy (Seno and Ima 2022) is utilized for training.

* 1. Results

In this section, first, we compare the convergence of different RL algorithms for this specific problem. Then using the best algorithm, we investigate how the different subsets of the data sets can affect the performance of RL agent. Finally, we compare the action and observations of one golden-batch experiment with the optimal RL agent.

* + 1. Offline RL Algorithms Comparison

Figure 1.a illustrates the training curves for BCQ, CQL, IQL and BC algorithms. For the training procedure we used the same data set (recipe 1, see sec 4.2) for different algorithms and the training procedures were performed for 18k training steps. This figure indicates that CQL outperforms the other algorithm both in performance and stability during the learning step. It also indicates that BCQ converges **to** a suboptimal policy, while IQL and BC didn’t converge to a certain policy. Moreover, Figure 1.a indicates that CQL converges to the optimal policy after 6k training steps and starts to overfits the data after 7k training steps, indicating a sign to stop the training procedure.

* + 1. Dataset Effect on Agent's Performance

CQL does not impose any specific assumption on the data sets. Some studies suggest that by combining datasets of different simple tasks, the trained agent can aggregate those simple tasks and perform more complex tasks (Kumar, Agarwal, et al. 2022). In this section we test the hypothesis that the quality of the logged episodes can affect the performance of an RL agent design to perform a simple control task in a chemical reaction. For a fixed reward function, Figure 2 illustrates the reward distribution for different subsets of experiments, and the average reward for full episodes is summarized in Table 1. As it is shown in this figure, the distribution of the reward for the full data sets and the subset with only recipe 1 is almost the same and the difference in average reward for these data sets is about 8.1%.

Figure 1.b represents the learning curve for CQL with the same hyperparameters using different subsets of training data. As it is shown in this figure, using the full data set, the performance of the agent increases during first few steps of the training (training step less than 2k). Afterwards, the performance of the agent decreases by continuing the learning steps. This figure shows by removing recipe 2 (with longer feed time) from the data sets, the training procedure became stable and converge to optimal policies. Training using only golden batch can easily over fits the policy leading to sub optimal policy, while using a data set consisting of only recipe 1 illustrates a robust learning procedure.

|  |  |
| --- | --- |
| A graph showing the number of training steps  Description automatically generated(a) | (b) |

Figure 1. returned reward of the agent during the training procedure, (a) comparing different algorithm using recipe 1 and (b) comparing performance of CQL using different subset of data

|  |  |  |
| --- | --- | --- |
| A graph of blue bars  Description automatically generated with medium confidence(a) | A graph with numbers and lines  Description automatically generated(b) | A graph with numbers and lines  Description automatically generated(c) |

Figure 2. distribution of reward for (a) full experimental data set, (b) only experiments with recipe 1, and (c) for only golden-batch experiments.

Table 1. Summary of average reward of different datasets and the best performance of RL agent during training.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Full data set | Recipe 1 | Golden batches |
| Data set average reward | -26.5 | -24.5 | -6.8 | |
| CQL best reward (training step) | -13.4 (1k) | -9.9 (18k) | -7.5 (2.6k) | |

* + 1. Performance Analysis of the agent

To analyze the performance of the trained agent, we initialize the simulation model based on one of the golden-baches experiments, let the agent interact with the environment, and compare the actions and observations of RL agent with experimental data as baseline. As shown in Figure 3, the actions of the RL agent have good agreement with the actions in the data set, indicating that the agent learnt the optimal policy from the experiments.

Figure 4 compares the states during the experiments. The solid lines are the experimental values (for one golden-batch experiment), and dash lines represent the simulation values controlled by RL agent. Table 2 summarized the correlation and errors between the golden-batch experiment and RL agent. As it is represented in this table, the main deviation is related to the temperature. It is mainly due to gap between simulation and real world (the accuracy off heat transfer equations is not as good as mass balance) and the fact that in experiments the temperature are changed discretely but the action space for the agent is continuous. Additionally, we analysed how measurement noise affects the agent’s performance. We used a random Gaussian noise model added to only measurement values as a percentage of the original value. Experiments show that increasing noise levels to 8% reduces mean reward by approximately 3.25%, while 10% noise can lead to a 15.3% reduction.

A graph of a temperature

Description automatically generated with medium confidence

Figure 3. comparison of actions: (a) flow rate of the feed, and (b) setpoint temperature of the heat bath.

A graph of a normalized temperature

Description automatically generated

Figure 4. comparison of observations: (a) concentration of component A, and (b) the mixture temperature.

Table 2. correlation coefficient and mean squared error (MSE) of main state elements compared to one of golden-batch experiments.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | Temperature |
|  | 0.894 | 0.999 | 0.705 |
|  | 0.004 | 0.003 | 5.711 |

* 1. Conclusions

This research demonstrates the application of offline RL algorithms to control a chemical reaction and reproduce the optimal batches. A dataset of 44 experiments, utilizing two recipes and various strategies, was used to train the RL agent through algorithms like BCQ, IQL, CQL, and BC. To investigate the performance of RL agent an ideal simulation model based on first principal heat and mass balance equations is developed to mimic the experimental setup. Using the same data set, CQL has the better performance in converging to the optimal policy while BC has the worth performance as there is no long-term index in this algorithm. Moreover, the data quality's impact was assessed by segmenting the datasets. Our investigations indicate that combining two recipes can reduce the performance of RL agent, significantly, while training on only golden-batch experiments can slightly improve the final performance of the agent but affects generalization. This paper showed the performance of offline RL algorithm in chemical industry. The future work will compare the performance offline RL agent with classical online methods. The reward function employed in this study was a simple function to follow predefined setpoints. The future work will design more complex reward function to optimize the process based on different objectives.

Acknowledgment

This work was performed within the CHAI imec ICON project. It was realized with the financial support of Flanders Innovation & Entrepreneurship (VLAIO, project no. HBC.2021.0671)

References

Sutton, Richard S, and Andrew G Barto. 2018. *Reinforcement Learning: An Introduction.* Cambridge, MA, USA: A Bradford Book.

Ma, Yan, Wenbo Zhu, Michael G Benton, and José Romagnoli. 2019. "Continuous control of a polymerization system with deep reinforcement learning." *Journal of Process Control* 75: 40-47.

Kumar, Aviral, Aurick Zhou, George Tucker, and Sergey Levine. 2020. "Conservative Q-Learning for Offline Reinforcement Learning." Edited by H Larochelle, M Ranzato, R Hadsell, M F Balcan and H Lin. *Advances in Neural Information Processing Systems* (Curran Associates, Inc.) 33: 1179-1191.

Fujimoto, Scott, David Meger, and Doina Precup. 2019. "Off-Policy Deep Reinforcement Learning without Exploration." Edited by Kamalika Chaudhuri and Ruslan Salakhutdinov. *Proceedings of Machine Learning Research.* PMLR. 2052-2062.

Kostrikov, Ilya, Ashvin Nair, and Sergey Levine. 2021. "Offline reinforcement learning with implicit q-learning." *arXiv preprint arXiv:2110.06169.*

Fujimoto, Scott, and Shixiang (Shane) Gu. 2021. "A Minimalist Approach to Offline Reinforcement Learning." *Advances in Neural Information Processing Systems.* Curran Associates, Inc. 34.

Brockman, Greg, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. *OpenAI Gym.*

Seno, Takuma, and Michita Ima. 2022. "d3rlpy: An Offline Deep Reinforcement Learning Library." *Journal of Machine Learning Research* 23: 1-20.

Levine, Sergey, Aviral Kumar, George Tucker, and Justin Fu. 2020. "Offline reinforcement learning: Tutorial, review, and perspectives on open problems." *arXiv preprint arXiv:2005.01643.*

Elmaz, Furkan, Ulderico Di Caprio, Min Wu, Yentl Wouters, Geert Van Der Vorst, Niels Vandervoort, Ali Anwar, M. Enis Leblebici, Peter Hellinckx, and Siegfried Mercelis. 2023. "Reinforcement learning-based approach for optimizing solvent-switch processes,." *Computers & Chemical Engineering* 176: 108310.

Kumar, Aviral, Rishabh Agarwal, Xinyang Geng, George Tucker, and Sergey Levine. 2022. "Offline q-learning on diverse multi-task data both scales and generalizes." *arXiv preprint arXiv:2211.15144.*