A Literature Review on Batch-Constrained Reinforcement Learning

Knut Roar Lende (4785525), Reinier Vos (4663160), Rinto de Vries (4881699)

Delft University of Technology
March 31, 2022

Abstract

Data gathering can be the most limiting part of reinforcement learning. A solution for this is to collect data on state-action pairs in exploratory setting, after which algorithms learn in an offline setting. This setup has paved the way for standard off-policy deep reinforcement learning algorithms such as DDPG and DQN. However, these algorithms are faced with a phenomenon known as the extrapolation error. This phenomenon can cause significant performance deterioration in cases of proposals for state-action pairs not contained in the available data. Hence, algorithms proposing constraining of the state-action space to the known regions are introduced. This literature review focuses on addressing this extrapolation error and evaluates the batch-constrained algorithms which intend to combat it. Batch-constrained Q-learning [1] is evaluated first, after which three competing methods (BEAR [2], BRAC [3] & MORel [4]) are introduced and evaluated. Ultimately, a comparison proposes a qualitative ranking of these algorithms and shows all competing methods outperform their predecessor BCRL. Further comparison between these methods shows MORel narrowly dominates, closely contending with BRAC.

Keywords: Batch-constrained reinforcement learning, offline reinforcement learning, imitation learning, off-policy, extrapolation error, BCQ, BEAR, BRAC, MORel

I. INTRODUCTION

In theory, off-policy Reinforcement Learning (RL) algorithms allow learning a policy from data gathered by other policies. However, in practice, true off-policy learning suffers from a fundamental issue called extrapolation error which can be one of the causes for significant losses in performance due to a distribution mismatch between data induced by the policy and that contained in the batch. Consequently, it might become impossible to adequately learn value functions for a selected policy not part of the batch [1]. Several strategies exist to minimize the effects of this issue and obtain robust performance in an true off-policy learning setting. This literature will define the nature of true off-policy learning. Furthermore, it attempts to explain the extrapolation error in a batch setting and evaluating batch-constrained algorithms that overcome the extrapolation error. This algorithm will also be compared to competing methods.

This literature review will be presented in the following form: section II provides the reader with some background in deep reinforcement learning. section III discusses the causes and effects of the extrapolation error in detail. section IV explains the BCQ algorithm (batch-constrained deep Q-learning) works. section V provides the reader with four competing methods. In section VI these four algorithms will be compared to BCQ, followed by a discussion of the results. section VII

II. BACKGROUND

The target of RL is to develop a strategy for an agent to maximize a reward in an environment. The strategy takes the form of a policy, π. RL-problems are typically formulated as Finite Markov Decision Processes (MDPs), that is, the state space $\mathcal{S}$ and action space $\mathcal{A}$ are both finite, and the probability distribution of arriving in any new state s' and the corresponding reward r depend solely on the previous state-action pair (s, a). [5]

The majority of RL algorithms learn and improve their policies by actively interacting with the environment [1]. This is referred to as online learning, or active data collection, where the agent runs multiple episodes in the environment, attempting different actions, and collecting rewards every episode.

The state-action value function, $Q(s, a)$, outputs the value of taking action a in state s. The value is measured by the expected return of the state-action pair [1]. Q-learning and deep Q-learning learns an estimate of the state-action value function through a backup-diagram and a neural network, respectively, and has as optimal policy to take the action with the highest value at every time step [5].

A distinction can be drawn between on- and off-policy learning algorithms, which are different ways of ensuring continued exploration of the environment. On-policy learning seeks to improve the decision making policy exploring the environment, while off-policy learning attempts to optimize a target policy π from data gathered under a behavioral policy μ. [5]
Fujimoto et al. [1] ascribe the limited application of RL in real world problems to the necessity of active data collection. The problem is attempted solved through batch-RL, where a policy is optimized based on a dataset collected under a (known) behavioral policy. Active data collection is thus prohibited.
Another sub-field in RL is imitation learning, where the algorithm have access to multiple episodes gathered under an expert behavioral policy μ[6]. Fujimoto et al. [1] argues that although there are similarities in the task formulations, imitation learning is different from batch-RL as it requires the data gathering policy to be close to some optimum, as it either aims to learn the optimal policy or replicate the resulting behavior [7–9]. A counter-example is the algorithm developed by Gao et al. [10], which can learn discrete actions from non-optimal demonstrations. Similar to the other imitation learning methods, it does require further online learning or continued access to an expert acting as a critic [1].

III. EXTRAPOLATION ERROR

In an off-policy value learning setting it is possible for the system to update its target policy to an action which is either improbable or impossible. This occurs because the value estimates, $Q(s, a)$, for a certain known state s (from the data set) are extrapolated to an unknown state-action pair (s', a') under the assumption that the Q function can be approximated adequately in adjacent state-action pairs (outside the data set). In the event that this assumption does not hold the system is faced with the extrapolation error.
Conceptually, the extrapolation error can be attributed to a value mismatch between the true state-action pairs and those available in the data set, see Equation 1. In this case subscript$Q_{(B)}$ represents the Q value as associated in the batch, while the other Q does not have this restriction.

$$\epsilon_{MDP}(s, a) = Q^{\pi}(s, a) - Q^{\pi}_{(B)}(s, a) \tag{1}$$

Potential specific origins for this mismatch can be identified [1]. First of all, for a proposed state-action pair the data might be absent. In this case the amount of available similar data in adjacent state-action pairs governs the extent of the error.
Secondly, the transition dynamics governed by the Bellman operator can be significantly biased. To some extent this bias always exists, as a finite size batch of transitions is used to approximate a true stochastic MDP. This is impossible to avoid, as infinite state-action visitation is impossible. This approximation is given in Equation 2. Where T^{π} is the bellman operator and $\mathcal{B}$ represents the batch distribution.

$$T^{\pi}Q(s, a) \approx \mathbb{E}_{s' \sim \mathcal{B}}[r + \gamma Q(s^{'}, \pi(s^{'}))] \tag{2}$$

Thirdly, the distribution of the current training batch might not resemble the distribution under the current policy close enough. As a result the batch sampled transitions in deep Q-learning might be inadequate. This is known as a training mismatch and can occur even if the size of the data set is substantial. Finally, even in cases with sufficient and adequate data the extrapolation error can be caused by the phenomenon of catastrophic forgetting [11],[12]. Catastrophic learning occurs whenever a network drastically underfits on previously seen data due to an inconsistency in distribution with currently observed samples.

The impact of the extrapolation error on Deep Q-learning algorithms [13] which have been labeled as off policy [14] such as DDPG [15] is underestimated. Previous research on the DDPG algorithm shows that its performance does not deteriorate as rapidly as the theorems on extrapolation error predict. [1] suggests that many of these 'off-policy' algorithms are in fact near-on-policy due to high correlation between the generated data sets and the current policy. Testing this algorithm in a truly off-policy setting by comparing the performance of behavioral and off-policy agents in OpenAI gym's Hopper-v1 environment [16] [17] show that indeed the extrapolation error causes a significant deterioration in performance [1]. Specifically, this can be attributed to diverging behavior in the value estimates, $Q(s, a)$ which are highly unstable. It should be noted that due to the scale of the experiments performed, the specific causes of the extrapolation error (as described in the previous paragraph) cannot be confidently identified, yet they do show the effect it has on performance. Concluding, these results show that most Deep Q-learning algorithms so far break down in truly off-policy settings. This suggests the need for extrapolation error robust off-policy algorithms with stable behavior in the value estimation. The next sections describe algorithms which intend to combat this extrapolation error.

IV. BATCH-CONSTRAINED REINFORCEMENT LEARNING

Batch-Constrained Reinforcement Learning (BCRL) is a form of off-policy learning used to overcome extrapolation error especially when faced with a high-dimensional continuous action space [16], [17]. In order to address the extrapolation error, BCRL proposes only policies that visit known state-action pairs in the batch [1]. These types of

policies are considered to be batch-constrained.
One form of Batch-Constrained Reinforcement Learning is Batch-Constrained deep Q-learning(BCQ) which uses a deep neural network. BCRL is augmented with a Q-network to produce unbiased estimates using offline data sets. Furthermore, it bridges the gap between imitation learning and off-policy learning as it can learn either from expert guidance or offline data. However an important distinction Fujimoto et al. [1] state is that for any initial state that is part of the batch, BCQ is assured to surpass any behavioral policy. Effectively, this means BCQ outperforms imitation learning. Moreover, it is achieved without any other condition on state-action visitation, besides adherence to the batch. This is in contradiction with Q-learning.

BCQ consists of four parametrized networks [1]. First of all, a generative model $G_\omega(s)$ which uses a conditional variational auto-encoder (VAE) [18], [19], to encapsulate the distribution by transformation of an underlying latent space. From this latent space policies can be drawn. If a single policy is drawn, the algorithm resembles imitation learning as the drawn policy will be followed. On the other hand, if the number of samples that are drawn increases, the algorithm will ultimately converge to Q learning as it will select one of the sampled policies based on their values. Secondly, the perturbation model $\xi_\theta(s, a)$ restricts the generative sample space to a desired range in order to increase diversity of policies within this range. Without this adjustment, the desired range might not contain diverse enough samples. Finally, two Q-networks $Q_{\theta 1}(s, a)$ and $Q_{\theta 2}(s, a)$ are included in the framework. These two Q-networks are used in the form of a Clipped Double Q-learning which is used to prevent overestimation of values. This is done by choosing the lowest value between the two networks. Furthermore, the value function $Q(s, a)$ can be optimized by training the perturbation model as well [20].

V. COMPETING METHODS

A. BEAR

Kumar et al.[2] presents the *Bootstrapping Error Accumulation Reduction* (BEAR) Q-Learning algorithm. It aims to solve the batch-constrained RL problem by requiring the state-action pairs executed under the learned policy to be close to the state-action pairs present in the dataset. An upper bound on the accumulated error under the restricted backup operator is further provided. The authors deem the constraint on the learned policy of BCQ to lie within the support of the behavioral policy used to gather the data too strict, and points out that regardless of the size of the dataset, the BCQ algorithm is unable to learn a strong policy if the behavior policy is close to uniform [2].

B. BRAC

Behavior Regularized Actor Critic (BRAC) is a general algorithmic framework which generalizes existing approaches to solve the offline reinforcement learning problem (the extrapolation error) by regularizing to the behavior policy. This is done by a couple of design choices: The type of regularization (value penalty, vp, or policy regularization, pr) and the type of divergence function between the learned policy and the behavior policy. By selecting specific design choices BCQ and BEAR can be recreated in the BRAC framework.

C. MORel

The model-based offline reinforcement learning algorithm is proposed by [4]. Prior to MORel, offline learning was mostly restricted to model-free RL algorithms [1],[2],[21–25]. MORel first learns a pessimistic MDP (P-MDP) using the offline data set, after which a near-optimal policy is learned on this P-MDP. Pessimism is incorporated using the unknown state-action detector (USAD). The performance in the P-MDP provides an approximate lower bound for the performance in the real MDP. Similar to BCQ, MORel penalizes policies visiting unknown states to reduce the extrapolation error. However, by using the USAD MORel avoids regularization of the policies through the data logging policy [2],[22],[23]. Consequently, MORel can be less conservative than model free algorithms. Furthermore, novel incorporation of generalization together with pessimism enables MORel to obtain policy improvements in states excluded in the available data set [4].

VI. COMPARISON AND DISCUSSION

A. BCQ

BCQ is compared against DQN [13] and DDPG [15] in a reproducible setting. Four learning cases are considered; learning of a final buffer, learning concurrently through the behavioral buffer and learning by expert imitation. A final imperfect demonstration setting is proposed by imposing noise on expert behavioral policy to evaluate BCQ robustness [1]. In all of these 4 settings, BCQ is able to match (and outperform) the pioneer algorithms. Without introducing additional sets of hyper parameters, BCQ is shown to be able to utilized for imitation and off-policy learning. In addition, BCQ seems to be mitigate the extrapolation error as a stable value function near off-policy samples is observed [1]. Finally, it is shown that BCQ is both robust to the imposed noise in the imperfect setting and able to learn in significantly fewer iterations.

B. BEAR

Kumar et al. [2] compares the performance of the BEAR algorithm to BCQ, Naive RL, and Behavioural Cloning on data sampled under an optimal, mediocre and random policies in four different benchmark tests, all part of the D4RL dataset. Bear is demonstrated capable of learning from data collected under the random policy. It further significantly outperforms the mediocre policy, and is able to match the optimal policy. BCQ is found to perform similarly to BC on the random and optimal data. It outperforms BC on data gathered under a mediocre policy, but is consistently falling short of the BEAR algorithm. [2]

C. BRAC

In the paper it is shown that the best performing version of BRAC (Value penalty with KL divergence to vanilla SAC) outperforms BCQ and BEAR on several benchmark tests, as can be seen in Figure 1. In these benchmark test an agent learns to walk with control of several torques. Each benchmark test has its own to be controlled shape (e.g. an ant). The plot shows performance for several datasets and cumulative reward against training steps, both averaged over multiple runs, for each benchmark. It can be seen that the best performing version of BRAC converges faster and performs better overall. Additionally, the experiments show that many complex training techniques used in BCQ and BEAR (e.g. adaptive regularization coefficients) are not necessary for maximum performance. The authors rather stress the importance of tuning the hyperparameters. [3]

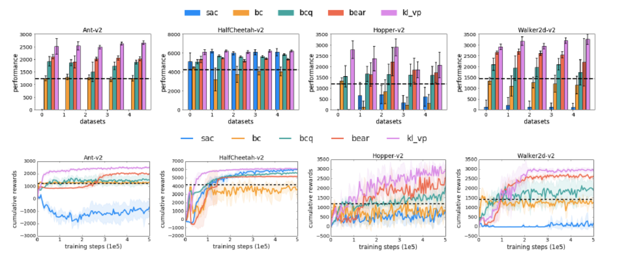
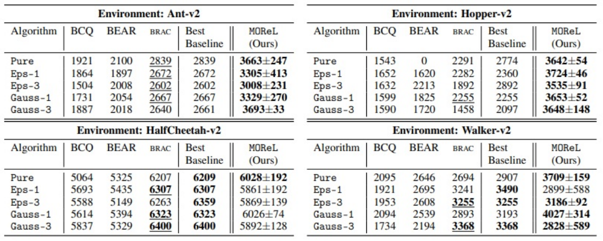


Fig. 1 "Comparing value penalty with KL divergence (kl_vp) to vanilla SAC, behavior cloning (bc), BCQ and BEAR. Bottom row shows sampled training curves with 1 out of the 5 datasets."[3]

D. MORel

MORel is first compared to naive Model-based RL (MBRL) to highlight the importance of the pessimism property with regard to over-exploitation. MORel does not show the same performance degradation after continuous policy improvement [4], unlike naive MBRL. Instead it shows that MORel results in a more stable and monotonic learning process, which can be attributed to the pessimism property [4]. Furthermore, out of the 20 experiments performed in 4 testing environments, which are all part of the D4RL dataset, MORel is shown to have superior performance in 12 cases when compared to naive MBRL and the other offline algorithms BCQ [1], BEAR [2], BRAC [3]. Figure 2 shows the results of these tests. The table presents the values obtain by the policies for the algorithms in the 4 environments. A best case scenario is presented for all algorithms except for MORel which shows an additional uncertainty margin to substantiate the comparison.



Environment: Ant-v2

Algorithm	BCQ	BEAR	BRAC	Best Baseline	MOReL (Ours)
Pure	1921	2100	2839	2839	**3663±247**
Eps-1	1864	1897	2672	2672	**3305±413**
Eps-3	1504	2008	2602	2602	**3008±231**
Gauss-1	1731	2054	2667	2667	**3329±270**
Gauss-3	1887	2018	2640	2661	**3693±33**

Environment: Hopper-v2

Algorithm	BCQ	BEAR	BRAC	Best Baseline	MOReL (Ours)
Pure	1543	0	2291	2774	**3642±54**
Eps-1	1652	1620	2282	2360	**3724±46**
Eps-3	1632	2213	1892	2892	**3535±91**
Gauss-1	1599	1825	2255	2255	**3653±52**
Gauss-3	1590	1720	1458	2097	**3648±148**

Environment: HalfCheetah-v2

Algorithm	BCQ	BEAR	BRAC	Best Baseline	MOReL (Ours)
Pure	5064	5325	6207	**6209**	**6028±192**
Eps-1	5693	5435	**6307**	**6307**	5861±192
Eps-3	5588	5149	6263	**6359**	5869±139
Gauss-1	5614	5394	**6323**	**6323**	6026±74
Gauss-3	5837	5329	**6400**	**6400**	5892±128

Environment: Walker-v2

Algorithm	BCQ	BEAR	BRAC	Best Baseline	MOReL (Ours)
Pure	2095	2646	2694	2907	**3709±159**
Eps-1	1921	2695	3241	**3490**	2899±588
Eps-3	1953	2608	**3255**	**3255**	**3186±92**
Gauss-1	2094	2539	2893	3193	**4027±314**
Gauss-3	1734	2194	**3368**	**3368**	**2828±589**

Fig. 2 Comparative table consisting of the results of the 20 learning of locomotive gaits tests under various levels of epsilon-greedy ('eps') and Gaussian noise ('Gauss') for the noisy variant of the policies [4].

E. Discussion

As shown in both Figure 2 and Figure 1, BEAR and the best performing version of BRAC outperform BCQ using the same benchmark test from the D4RL dataset. It can also be noted that the best performing version of BRAC (not BEAR) outperforms BEAR on every single benchmark test. Lastly, Figure 2 shows that MORel outperforms BEAR and BCQ on all the used benchmarks, but is not strictly better than BRAC (over 20 experiments, BRAC won eight and MORel won twelve). [4][3]. Based on these benchmark tests it can be concluded that MORel and BRAC perform better than BEAR, which performs better than BCQ. All in all, MORel narrowly dominates, but its close contention with BRAC indicates no clear proclamation of supremacy.

VII. CONCLUSION

In this literature review the foundation upon which batch-constrained reinforcement learning is provided. After this, the extrapolation error is investigated further and algorithms which intend to combat it are proposed and compared. This review specifically rejects the 'off-policy' nature of pioneer RL algorithms such as DDPG and DQN and underlines the potential effect of the extrapolation error. This extrapolation error is accredited to absent data, biased transition dynamics and training data distribution mismatch. BCQ is proposed as the first algorithm that incorporates measures to combat value overestimation for unknown state-action pairs. Although BCQ might outperform the pioneer algorithms, it itself is outperformed by later proposals. Ultimately, it is concluded that that MORel and BRAC perform better than BEAR, which in turn performs better than BCQ.

References

[1] Fujimoto, S., Meger, D., and Precup, D., "Off-policy deep reinforcement learning without exploration," *International Conference on Machine Learning*, PMLR, 2019, pp. 2052–2062.

[2] Kumar, A., Fu, J., Tucker, G., and Levine, S., *Stabilizing Off-Policy Q-Learning via Bootstrapping Error Reduction*, chapter and pages.

[3] Wu, Y., Tucker, G., and Nachum, O., "Behavior Regularized Offline Reinforcement Learning," *CoRR*, Vol. abs/1911.11361, 2019. URL `http://arxiv.org/abs/1911.11361`.

[4] Kidambi, R., Rajeswaran, A., Netrapalli, P., and Joachims, T., "MOReL: Model-Based Offline Reinforcement Learning," *Advances in Neural Information Processing Systems*, Vol. 33, edited by H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Curran Associates, Inc., 2020, pp. 21810–21823. URL `https://proceedings.neurips.cc/paper/2020/file/f7efa4f864ae9b88d43527f4b14f750f-Paper.pdf`.

[5] Sutton, R. S., and Barto, A. G., *Reinforcement learning: An introduction*, MIT press, 2018.

[6] Hussein, A., Gaber, M. M., Elyan, E., and Jayne, C., "Imitation learning: A survey of learning methods," *ACM Computing Surveys (CSUR)*, Vol. 50, No. 2, 2017, pp. 1–35.

[7] Hester, T., Vecerik, M., Pietquin, O., Lanctot, M., Schaul, T., Piot, B., Horgan, D., Quan, J., Sendonaris, A., Osband, I., et al., "Deep q-learning from demonstrations," *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32, 2018.

[8] Sun, W., Bagnell, J. A., and Boots, B., "Truncated horizon policy search: Combining reinforcement learning & imitation learning," *arXiv preprint arXiv:1805.11240*, 2018.

[9] Chemali, J., and Lazaric, A., "Direct policy iteration with demonstrations," *Twenty-fourth international joint conference on artificial intelligence*, 2015.

[10] Gao, Y., Xu, H., Lin, J., Yu, F., Levine, S., and Darrell, T., "Reinforcement learning from imperfect demonstrations," *arXiv preprint arXiv:1802.05313*, 2018.

[11] McCloskey, M., and Cohen, N. J., "Catastrophic interference in connectionist networks: The sequential learning problem," *Psychology of learning and motivation*, Vol. 24, Elsevier, 1989, pp. 109–165.

[12] Goodfellow, I. J., Mirza, M., Xiao, D., Courville, A., and Bengio, Y., "An empirical investigation of catastrophic forgetting in gradient-based neural networks," *arXiv preprint arXiv:1312.6211*, 2013.

[13] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al., "Human-level control through deep reinforcement learning," *nature*, Vol. 518, No. 7540, 2015, pp. 529–533.

[14] Watkins, C. J. C. H., "Learning from delayed rewards," 1989.

[15] Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D., "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[16] Todorov, E., Erez, T., and Tassa, Y., "Mujoco: A physics engine for model-based control," *2012 IEEE/RSJ international conference on intelligent robots and systems*, IEEE, 2012, pp. 5026–5033.

[17] Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W., "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.

[18] Kingma, D. P., and Welling, M., "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[19] Sohn, K., Lee, H., and Yan, X., "Learning structured output representation using deep conditional generative models," *Advances in neural information processing systems*, Vol. 28, 2015.

[20] Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M., "Deterministic policy gradient algorithms," *International conference on machine learning*, PMLR, 2014, pp. 387–395.

[21] Agarwal, R., Schuurmans, D., and Norouzi, M., "Striving for simplicity in off-policy deep reinforcement learning," 2019.

[22] Wu, Y., Tucker, G., and Nachum, O., "Behavior regularized offline reinforcement learning," *arXiv preprint arXiv:1911.11361*, 2019.

[23] Jaques, N., Ghandeharioun, A., Shen, J. H., Ferguson, C., Lapedriza, A., Jones, N., Gu, S., and Picard, R., "Way off-policy batch deep reinforcement learning of implicit human preferences in dialog," *arXiv preprint arXiv:1907.00456*, 2019.

[24] Laroche, R., Trichelair, P., and Des Combes, R. T., "Safe policy improvement with baseline bootstrapping," *International Conference on Machine Learning*, PMLR, 2019, pp. 3652–3661.

[25] Nachum, O., Dai, B., Kostrikov, I., Chow, Y., Li, L., and Schuurmans, D., "Algaedice: Policy gradient from arbitrary experience," *arXiv preprint arXiv:1912.02074*, 2019.