Proceedings of Machine Learning Research vol 144:1–2, 2022

Supplementary information for the article: A Novel Data Augmentation Framework for Optimal State-Action Pairs

Dinesh Krishnamoorthy DKRISHNAMOORTHY@SEAS.HARVARD.EDU
Harvard John A. Paulson School of Engineering and Applied Sciences, Cambridge, MA, 02138.

1. Detailed discussion of the Numerical results

1.1. Direct policy approximation

In the numerical results, section, we used a grid-based sampling approach to sample the feasible state-space for the inverted pendulum example. In this case, we queried the expert to generate $N = 90$ samples using a sparse grid. We denote this base data set as $\mathcal{D}^0$. We further augment this data set with additional $M = 8958$ samples by using the proposed method around each sample point. This is denoted by $\mathcal{D}^+$. That is, we generate a total of 9040 data points using only 90 queries to the expert. We also generate the exact samples by querying the expert at the same points as all the augmented samples (i.e query the expert 9040 times). This densely sampled data set is denoted by $\mathcal{D}^{++}$

Using the data, we now fit a parametric function to approximate the policy, and we compare the closed-loop performance of the policy learned using the different data sets. In this paper, we use a neural network with five layers with 10 neurons in each layer. The neurons use $\tanh(\cdot)$ as the activation function. The neural network was trained using the `fitnet` function in `MATLAB v2020b` with Levenberg-Marquardt backpropagation. Note that the choice of the function approximator and its hyper-parameters are not the focus of this paper, and were chosen based on simple trial and error that is adequate to illustrate the effect of the proposed data augmentation framework.

We first use only the sparsely sampled data set $\mathcal{D}^0$ to train the function approximator. The closed-loop performance of the resulting approximate policy $\pi_{sparse}(x, \theta_0)$ is shown in Fig. 1 in gray, where it can be immediately seen that the data set $\mathcal{D}^0$ is not sufficient to learn a satisfactory policy function. The approximate policy learned using the proposed augmented data set $\mathcal{D}^0 \cup \mathcal{D}^+$ is denoted as $\pi_{aug}(x, \theta_1)$. The closed-loop performance of this policy is shown in Fig. 1 in red. Here it can be seen that the policy learned from the augmented data set is able to approximate the control policy sufficiently accurately, and provides almost identical results as the true MPC policy (shown in blue). We also compare the performance of the policy $\pi_{dense}(x, \theta_2)$ learned using the densely sampled data set $\mathcal{D}^0 \cup \mathcal{D}^{++}$, and the corresponding closed-loop performance is shown in Fig. 1 (in green).

From Table 1 in the article, it can be seen that using a base data set of $\mathcal{D}^0$ with $N = 90$ samples, we can augment $M = 8858$ additional samples using only a fraction of time, and the learned policy provides near identical performance as the optimal policy as well as the policy learned from the computationally expensive densely sampled data set.

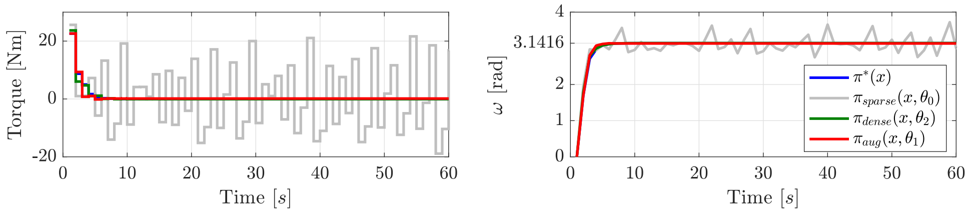


Figure 1: Closed loop performance of the approximate polices trained using the sparsely sampled data set, densely sampled data set, the proposed augmented data set.

1.2. Direct policy approximation with interactive expert

The proposed data augmentation framework is applied in the context of imitation learning with interactive expert on the same inverted pendulum example. The objective here is to learn a policy function that would bring the pendulum from rest position ($\omega = \dot{\omega} = 0$) to its inverted position ($\omega = 3.14, \dot{\omega} = 0$).

Task Learn to take a pendulum from rest to its inverted position.

Initial policy Linear policy $u = -Kx$ with a K is not able to accomplish the task.

Learning framework We first start with no prior expert demonstrations, but with an initial policy that is rolled out on the system. For this, we chose a linear policy $u = -11\omega - 7\dot{\omega} + 35$. For each state visited in the rollout, the expert provides feedback on what the expert would have done in these states. Around each expert feedback, we further augment 25 samples using the proposed data augmentation framework.

We first learn a policy function using only the expert feedback. To demonstrate the advantage of augmenting additional samples inferred from the expert feedback, we then learn a policy function based on the expert feedback and the augmented samples. For both these policies, we choose a generalized regression neural network that was trained using the `newgrnn` function in `MATLAB v2020b`. The newly obtained approximate policy is then rolled out and this procedure is repeated for 25 times.

Fig. 2 left subplots shows the states that are visited in each rollout, where the expert provides feedback (shown in red circles). As it can be seen, even after 25 rollouts the number of demonstrations in the state-space is not much, which can affect the quality of the policy learned using only the expert feedback. Whereas, by augmenting additional samples around each expert feedback, we can cheaply generate several data samples covering a wider state-space than just what was visited in the rollouts. This is shown in black dots in Fig. 2 right subplots. (From the 3D plot, one can already visually recognize the shape of the optimal solution manifold when we augment additional data samples).

The performance of the policy learned using only the expert feedback or using the expert and the augmented feedback for all the 25 rollouts are shown in Fig. 3. Here it can be seen that by augmenting additional samples, we are able to better learn the policy, and hence learn to control the inverted pendulum better than if we only use the expert feedback. This is because, after each rollout, we not only learn from the expert feedback, but also infer what the expert would have done in other states not visited in the current rollout.

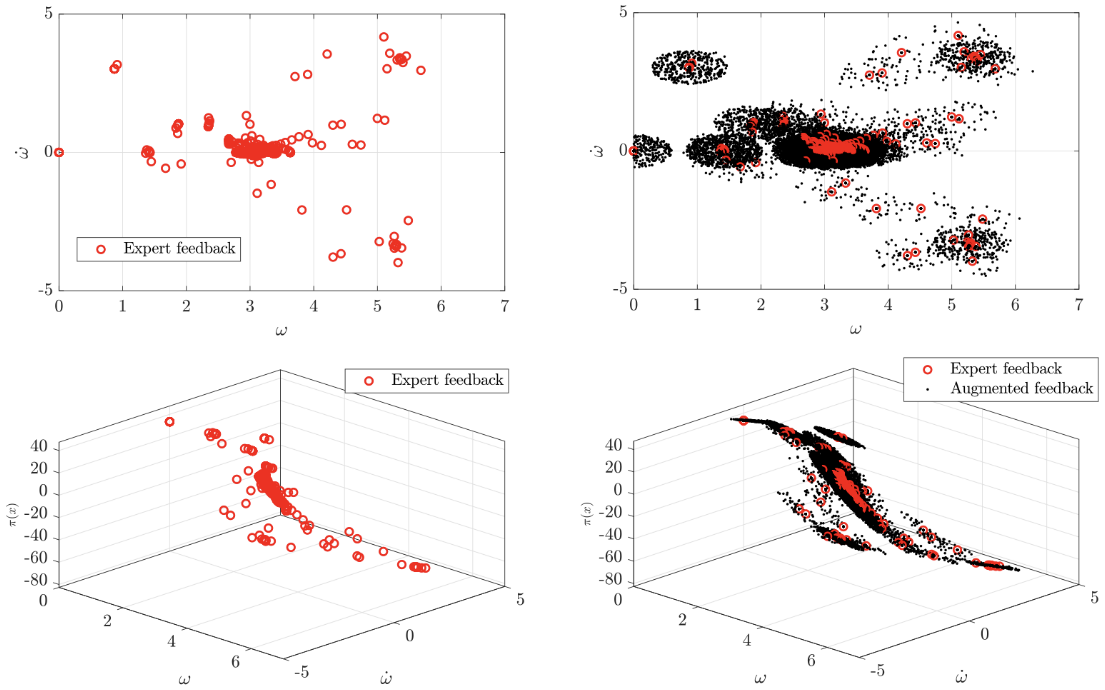
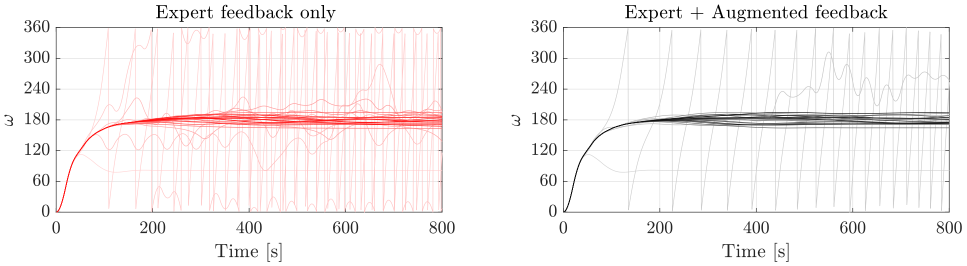


Figure 2: Closed loop performance over 15 rollouts when using only the expert feedback (shown in red), and using botht he expert and augmented feedback (shown in black). The bold lines shows the performance at the 15^{th} rollout.



Rollout #	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Expert only	✗	✗	✗	✗	✗	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓
Expert + Aug	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Figure 3: Closed-loop performance of all the rollouts using only expert feedback (left subplot) and using expert and augmented feedback (right subplot). Table shows whether the task in a particular rollout was successful or not.