How to Leverage Unlabeled Data in Offline RL

Tianhe Yu, Aviral Jumar, Yevgen Chebotar, Karol Hausman, Chelsea Finn, Sergey Levine

ICML 2022

Presented by: William Loh

David R. Cheriton School of Computer Science
University of Waterloo
wmloh@uwaterloo.ca

April 3, 2023

Outline

1 Introduction

2 Unlabeled Data Sharing (UDS)

3 Optimality and Implementation of UDS

4 Experimental Results

Motivation

- Targeted sampling in offline reinforcement learning is difficult
- In some domains, rewards has to be labelled by humans (especially in robotics)
- Data without reward annotations are relatively abundant

Question 1: *Can unlabeled data improve performance?*

Question 2: *How to incorporate unlabeled data into offline reinforcement learning training?*

Motivating Example

- **Task of interest**: Robot cutting an onion
- **Problem**: Relatively small labeled dataset on robots cutting an onion
- **Prior dataset**: Lots of data on robots cutting an onion without reward annotations, as well as plenty of data on picking up onions and chopping a carrot

Question 1: *Which prior dataset (if any) should be included for the new task?*
Question 2: *How should the reward labels of the prior data be determined for learning a new task?*

Related Works on Offline RL + Unlabeled Data

- Using all prior labeled data – only applicable to structurally, highly similar tasks
- Label propagation for rewards – requires a learned classifier and adds complexity to the pipeline
- Multi-task data sharing – requires access to the functional form of the reward for relabeling or limited to goal-conditioned settings

Illustration

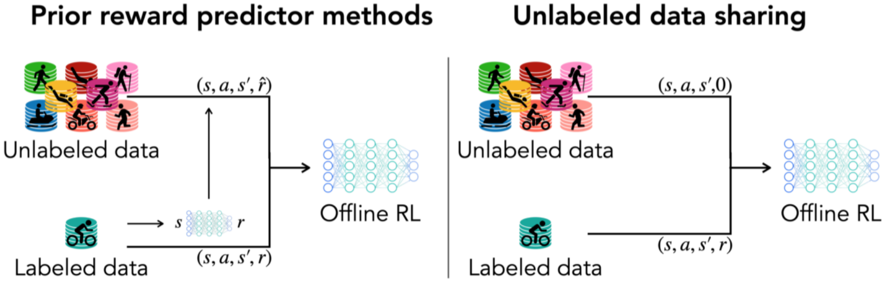

Figure 1: Comparison of methods on unlabeled data

Objective

Investigate the efficacy of labeling unlabeled data with reward of zero[1] *in various cases.*

- *without needing access to a functional form of rewards*
- *without additional modeling and learning*

[1] Set reward to the minimum reward in the dataset, or without loss of generality, zero via rescaling

Notation

- $\mathcal{D}_L = \{(s, a, s', r)\}$ is the **labeled dataset**
- $\mathcal{D}_U = \{(s, a, s', 0)\}$ is the **unlabeled dataset**
- π_β is the **behaviour policy** in the static dataset
- d^π is the **state-action marginal** of policy π
- J is the **objective function**

Outline

1 Introduction

2 Unlabeled Data Sharing (UDS)

3 Optimality and Implementation of UDS

4 Experimental Results

Formulation

Unlabeled data sharing (UDS) assigns the lowest possible reward to all transitions in $\mathcal{D}_U$.

- Claim that this strategy works in theory and practice *under certain conditions*
- UDS uses a combined dataset $\mathcal{D}^{\text{eff}} := \mathcal{D}_L \cup \mathcal{D}_U$

Implications of UDS in Offline Setting (Part 1)

Reward Bias

- Suboptimality due to using incorrect reward (since we're using reward of 0 for all unlabeled data)

$$\mathrm{RewardBias}(\pi^*_{\mathrm{UDS}}, \pi^{\mathrm{eff}}_{\beta}) = \frac{1}{1-\gamma}\sum_{s,a} \underbrace{\Delta(d^{\pi^{\mathrm{eff}}_{\beta}}, d^{\pi^*_{\mathrm{UDS}}})}_{\text{statistical distance}} \cdot (1 - \underbrace{f(s,a)}_{\text{ratio of labeled data}}) \cdot \underbrace{r(s,a)}_{\text{true reward}}$$

where

$$f(s,a) = \frac{|\mathcal{D}_L(s,a)|}{|\mathcal{D}^{\mathrm{eff}}(s,a)|}, \qquad \Delta(d^{\pi^{\mathrm{eff}}_{\beta}}, d^{\pi^*_{\mathrm{UDS}}}) = d^{\pi^{\mathrm{eff}}_{\beta}}(s,a) - d^{\pi^*_{\mathrm{UDS}}}(s,a)$$

Implications of UDS in Offline Setting (Part 2)

Sampling Error

- Epistemic error incurred from the lack of data (taken from a paper on multi-task offline RL)

$$\text{SamplingError}(\pi^*_{\text{UDS}}, \pi^{\text{eff}}_{\beta})$$
$$= \mathcal{O}\left(\frac{\gamma}{(1-\gamma)^2}\right) \mathbb{E}_{s,a\sim\hat{d}^{\pi}} \left[\sqrt{\frac{D_{\text{CQL}}(\pi^*_{\text{UDS}}, \pi^{\text{eff}}_{\beta})(s)}{|\mathcal{D}^{\text{eff}}(s)|}}\right]$$

where D_{CQL} is the statistical distance under conservative Q-learning.

Implications of UDS in Offline Setting (Part 3)

Policy Improvement

- Performance improvement induced by the transitions in $\mathcal{D}^{\text{eff}}$ that occurs as a result of offline RL

$$\text{PolicyImprov}(\pi^*_{\text{UDS}}, \pi^{\text{eff}}_{\beta}) = \frac{\alpha}{1-\gamma} D(\pi^*_{\text{UDS}}, \pi^{\text{eff}}_{\beta})$$

where D is a statistical distance.

Theorem I

Policy improvement guarantee for UDS

Theorem

Let π^*_{UDS} denote the policy learned by UDS and $\pi_\beta^{eff}(a|s)$ denote the behaviour policy for the combined dataset $\mathcal{D}^{eff}$. Then with high probability of at least $1-\delta$, π^*_{UDS} is a safe policy improvement over π_β^{eff}.

$$J(\pi^*_{UDS}) \geq J(\pi_\beta^{eff}) - \zeta_{err} + PolicyImprov(\pi^*_{UDS}, \pi_\beta^{eff})$$

where $\zeta_{err} = RewardBias(\pi^*_{UDS}, \pi_\beta^{eff}) + SamplingError(\pi^*_{UDS}, \pi_\beta^{eff})$.

Implications of Theorem I

$$\text{RewardBias}(\pi^*_{\text{UDS}}, \pi^{\text{eff}}_{\beta}) = \frac{1}{1-\gamma} \sum_{s,a} \Delta(d^{\pi^{\text{eff}}_{\beta}}, d^{\pi^*_{\text{UDS}}}) \cdot (1 - f(s,a)) \cdot r(s,a)$$

$$\text{SamplingError}(\pi^*_{\text{UDS}}, \pi^{\text{eff}}_{\beta}) = \mathcal{O}\left(\frac{\gamma}{(1-\gamma)^2}\right) \mathbb{E}_{s,a\sim\hat{d}^{\pi}} \left[\sqrt{\frac{D_{\text{CQL}}(\pi^*_{\text{UDS}}, \pi^{\text{eff}}_{\beta})(s)}{|\mathcal{D}^{\text{eff}}(s)|}}\right]$$

- Notice that in the error term ζ_{err}, the size of the unlabeled dataset $|\mathcal{D}_U|$ has an opposing effect on RewardBias and SamplingError
  - As $|\mathcal{D}_U|$ proportionately increases, the ratio of labeled data $f(s,a)$ decreases so the RewardBias increases
  - As $|\mathcal{D}_U|$ increases, the overall effective dataset size $|\mathcal{D}^{\text{eff}}|$ increases so the SamplingError decreases

Analysis of Trade-offs (Case 1)

Case 1 – *unlabeled data is distributed identically as labeled data*

- Large amount of offline data is available but only a limited uniformly sampled fraction is annotated with rewards
- This means that RewardBias is proportional to the sum of difference of performance (overall rewards) in the empirical MDP
- Learned policy in offline RL π^*_{UDS} improve over the effective behaviour policy π^{eff}_{β} so the RewardBias will be negative
- SamplingError will also decrease due to more data
- UDS improves performance without incurring additional cost due to the wrong reward

Analysis of Trade-offs (Case 2)

Case 2 – *Low true reward of the unlabeled dataset*

- When the unlabeled dataset in reality does not contain high true rewards, and we annotated them with a reward of 0, this does not incur much RewardBias
- We still get the benefits of implicitly or explicitly learning transitions in unlabeled dataset so it reduces SamplingError

Analysis of Trade-offs (Case 3)

Case 3 – large unlabeled datasets for long-horizon tasks

- In long-horizon tasks, $H := \frac{1}{1-\gamma}$ is large, and it affects RewardBias and SamplingError in different rate of growth
    - RewardBias grows linearly with H while SamplingError grows quadratically with H
- For cases where $|\mathcal{D}^{\text{eff}}(s)| = \Omega(H^2)|\mathcal{D}_L(s)|$, the overall ζ_{err} is asymptotically unchanged, while having more data for PolicyImprov

Discussion of Comparison with Reward Prediction (Part 1)

The general expression for reward bias is

$$\mathrm{RewardBias}(\pi, \pi_\beta^{\mathrm{eff}}) = \frac{1}{1-\gamma} \sum_{s,a} \Delta(\hat{d}^{\pi_\beta^{\mathrm{eff}}}, \hat{d}^{\pi}) \cdot \Delta r(s,a)$$

where $\Delta r(s,a)$ is the error in the reward applied to the unlabeled data.

In UDS, $\Delta r(s,a) = r(s,a) - 0$ but in a reward prediction method, it would be $\Delta r(s,a) = r(s,a) - \hat{r}(s,a)$. Note that $r(s,a) \geq 0$ since without loss of generality, 0 is the minimum reward.

Discussion of Comparison with Reward Prediction (Part 2)

Unlabeled Data Sharing

- $\Delta r(s, a) = r(s, a) \geq 0$ for all (s, a)
- Whenever $\hat{d}^{\pi_\beta^{\text{eff}}}(s, a) < \hat{d}^{\pi}(s, a)$, i.e. (s, a) appearing more frequently under the learned policy than the effective behaviour policy, this might reduce the sub-optimality from reward bias
- Intuitively can be seen as inducing a conservative behaviour on unlabeled data

Discussion of Comparison with Reward Prediction (Part 3)

Reward Prediction

- $\Delta r(s, a) = r(s, a) - \hat{r}(s, a)$ may not be positive on all (s, a) pairs and hence may incur reward bias
- Policy optimization seeks out policies that maximize $\hat{d}^{\pi}(s, a)$ on (s, a) with high rewards so $\Delta(\hat{d}^{\pi_{\beta}^{\text{eff}}}, \hat{d}^{\pi}) < 0$
- Reward prediction models tend to be biased towards out-of-distribution (OOD) action so $r(s, a) < \hat{r}(s, a)$ and hence $\Delta r(s, a) < 0$

Outline

1 Introduction

2 Unlabeled Data Sharing (UDS)

3 Optimality and Implementation of UDS

4 Experimental Results

Controlling and Optimizing Trade-offs

From the theoretical analysis, UDS only has benefits on selected conditions. To harness its potential and reduce sub-optimality induced by reward bias, we have to preferentially reweight transitions in $\mathcal{D}_L$.

- In existing literature, there are efforts to reduce *distributional shift*
- The scheme to reduce reward bias intuitively matches the scheme to reduce distributional shift

Theorem II

Optimized reward bias reduction

Theorem

The optimal effective behaviour policy that minimizes $RewardBias(\pi^*_{UDS}, \pi^{eff}_\beta)$ *satisfies*

$$d^{\pi^{eff}_\beta}(s,a) \propto \sqrt{d_L(s,a)d^\pi(s,a)}$$

where d^π *denotes the state-action marginal of policy* π *and* $d_L(s,a)$ *denotes the density of state-action pair* (s,a) *under the labeled dataset*

Implementation of Theorem II

- Theorem II essentially states that the effective behaviour policy π_{β}^{eff} must place mass on state-action tuples that are likely under the learned policy d^{π} and distribution induced by the label dataset d_L
- Computing state-action marginals can be challenging so authors utilize an existing method called **conservative data sharing** (CDS) to reweigh unlabeled data efficiently
  - CDS is meant for multi-task offline RL but they offer a practical solution on reweighting data from other sources efficiently
  - In this paper, the method of using UDS but with efficient reweighting of unlabeled data is called **UDS+CDS**

Outline

1 Introduction

2 Unlabeled Data Sharing (UDS)

3 Optimality and Implementation of UDS

4 Experimental Results

Evaluated Methods

- UDS
- UDS + CDS reweighting
- Variational inverse control with events (VICE) – learns a reward function through inverse RL on samples of desired goal states
- Recursive classification of examples (RCE) – learns a classifier to determine success or failure
- No sharing – using only labeled data (baseline)
- Reward prediction – naïve reward regressor

Single-task Domains

- 10,000 labeled transitions and 1,000,000 unlabeled transitions
- Unlabeled data is low-quality (i.e. low rewards and possibly irrelevant to target task)

Environment	Labeled data	Unlabeled data	CDS+UDS	UDS	No Sharing	Reward Pred.	VICE	RCE
D4RL hopper	expert	random	**81.5**	78.6	77.1	67.6	n/a	n/a
	expert	medium	**78.3**	64.4	77.1	51.7	n/a	n/a
D4RL AntMaze	expert	medium-play	82.6	**82.7**	17.2	0.0	0.0	0.0
	expert	large-play	**47.1**	33.1	0.7	0.0	0.0	0.0

Figure 2: Single task environments – Hopper and AntMaze

Multi-task Imaged-based Robotic Manipulation

- Use data from other tasks as $\mathcal{D}_U$ to train for a target task

Environment	Tasks	CDS+UDS	UDS	VICE	RCE	No Sharing	Reward Pred.
Meta-World	door open	**61.3%**±7.9%	51.9%±25.3%	0.0%±0.0%	0.0%±0.0%	14.5%±12.7%	0.0%±0.0%
	door close	54.0% ±42.5%	12.3%±27.6%	66.7%%±47.1%	0.0%±0.0%	4.0%±6.1%	**99.3%**±0.9%
	drawer open	**73.5%**±9.6%	61.8%±16.3%	0.0%±0.0%	0.0%±0.0%	16.0%±17.5%	13.3%±18.9%
	drawer close	99.3%±0.7%	**99.6%**±0.7%	19.3%±27.3%	2.7%±1.7%	99.0%±0.7%	50.3%±35.8%
	average	**71.2%** ± 11.3%	56.4%±12.8%	21.5%±0.7%	0.7%±0.4%	33.4%±8.3%	41.0%±11.9%
AntMaze	medium (3 tasks)	**31.5%**±3.0%	26.5%±9.1%	2.9%±1.0%	0.0%±0.0%	21.6%±7.1%	3.8%±3.8%
	large (7 tasks)	**18.4%**±6.1%	14.2%±3.9%	2.5%±1.1%	0.0%±0.0%	13.3% ± 8.6%	5.9%±4.1%

Figure 3: Multi-task robotic manipulation and navigation environment

Summary of Empirical Analysis

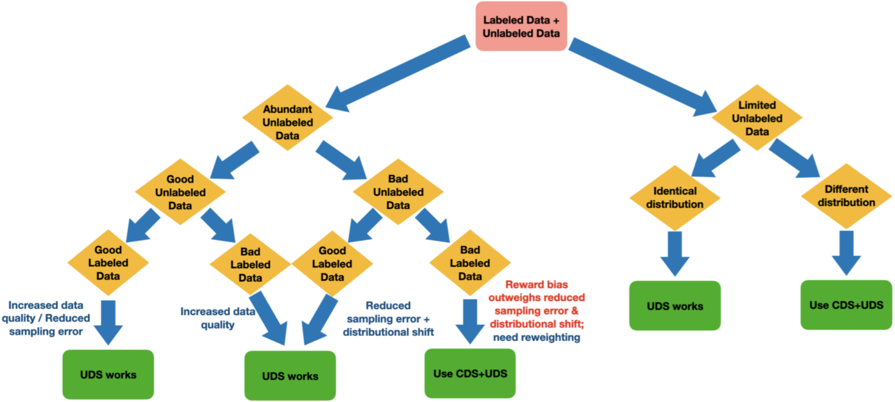

Figure 4: Conditions to use UDS and optimized reweighting (CDS) with UDS

Thank you!