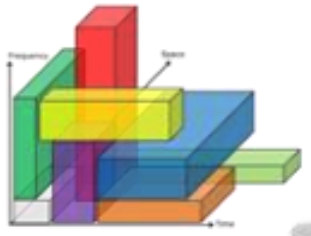


ANDRO

Constrained Offline Policy Optimization

Nicholas Polosky

International Conference on Machine Learning

July 2022

AGENDA

MOTIVATION

BACKGROUND

COPO

EXPERIMENTS

RESULTS

TAKEAWAYS

MOTIVATION

In offline reinforcement learning settings, we are provided with a static set of data collected from an unknown number of unknown policies. The goal is to learn a policy that performs well once deployed in the environment, using only the static data set [1].

Distributional shift occurs when the data distribution of the static data does not match that of the distribution which would be observed using the current learned policy [1].

Distributional shift can result in unreliable policy evaluation and is thus problematic for policy selection.

This problem is even more pertinent in safety-critical or constrained RL problems, in which, there are behaviors that an agent must strictly avoid (e.g. unsafe, illegal actions).

Accordingly, we introduce an offline policy optimization algorithm which accounts for distributional shift in constrained offline RL problems, titled Constrained Offline Policy Optimization (COPO).

Background

We work on problems represented using a Constrained Markov Decision Process (CMDP) [2], is defined using the tuple:

$$\langle S, A, P, R, C, \gamma, \mu \rangle$$

Representing the state space, action space, transition operator, reward function, cost function, discount parameter, and initial state distribution, respectively.

We work in the undiscounted ($\gamma = 1$), infinite horizon setting, where the value of a policy is defined as:

$$\rho(\pi) = \lim_{T \to \infty} \mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T} R(s_t, a_t) \mid s_0 \sim \mu, a_t \sim \pi(s_t), s_{t+1} \sim P(s_t, a_t) \forall t\right]$$

In the undiscounted setting, the Off-Policy Evaluation (OPE) problem may be represented using a linear program. This linear program is often called the Q-LP [3] and is provided below

$$\min_{\lambda, Q} \lambda \quad s.t.\, \mathrm{Q(s,a)} \geq R(s,a) + P^{\pi} Q(s,a) - \lambda$$

The dual linear program of the Q-LP is called the d-LP [3] and is written as:

$$\max_{d} \mathbb{E}_d R(s,a) \quad s.t.\, d(s,a) = P_*^{\pi} d(s,a), \qquad \sum d(s,a) = 1$$

In the above d is the normalized state-action visitation density, P^{π} is the transition operator under policy π, P_*^{π} is the adjoint policy transition operator, and λ is a normalizing variable.

Constrained Offline Policy Optimization

Constrained Projection

We set up the constrained projection problem by supposing that we had access to the state-action visitation density of a policy that maximized reward, we will call this the reward-optimal policy its visitation is the reward-optimal visitation denoted d^R.

The problem we want to solve is thus:

$$\min_{\pi} D(d^{\pi}, d^{R}) \quad \text{s.t. } \rho_C(\pi) \leq \beta$$

Where $\rho_C(\pi)$ is the cost-value of the policy, β is the cost budget, and D is a metric or pseudo-metric on policy space.

We can then take the Lagrangian, and expand the cost OPE term:

$$\min_{\pi} \max_{\lambda \geq 0} D(d^{\pi}, d^{R}) + \lambda \rho_C(\pi) - \lambda\beta$$

$$\min_{\pi} \max_{\lambda \geq 0} D(d^{\pi}, d^{R}) + \min_{d} \sum d(s,a)\lambda a C(s,a) - \lambda\beta \;\; s.t. \, d(s,a) = P_*^{\pi} d(s,a), \sum d(s,a) = 1$$

Adding the sum-to-one constraint to the objective and rearranging we get:

$$\min_{\pi} \max_{\lambda \geq 0, \nu} \min_{d} D(d, d^{R}) + \sum d(s,a)(\lambda C(s,a) + \nu) - (\lambda\beta + \nu) \;\; s.t. \, d(s,a) = P_*^{\pi} d(s,a)$$

Constrained Offline Policy Optimization

Constrained Projection

We can transform this problem using the following identity. Given the primal problem:

$$\min_x f(x) + g(Ax)$$

Fenchel-Rockafeller duality [4] yields the following dual problem:

$$\max_y -f_*(A_*y) - g_*(y)$$

Where A_* is the adjoint of A and a subscripted * represents the convex conjugate.

We set:

$$f(d) = D(d, d^R) + \sum d(s,a)(\lambda C(s,a) + \nu) - (\lambda\beta + \nu)$$

and

$$g(Ad) = \delta_0(Ad), \qquad A = I - P_*^\pi$$

With δ_0 as the 0-indicator function.

Changing the problem in this way allows for us to optimize π directly in the usual way, rather than through the state-action visitation variable d.

f-divergence distance:

$$D(d, d^R) = \mathbb{E}_{(s,a)\sim d^R}\left[f\left(\frac{d}{d^R}\right)\right]$$

f-divergence objective:

$$\max_{\substack{\lambda\geq 0,\nu,\\ Q_c:\mathcal{A}\times\mathcal{A}\to\mathbb{Z}}} -\alpha \mathop{\mathbb{E}}_{(s,a)\sim d^R}[\, f_*((P^\pi Q_c(s,a) - Q_c(s,a) - \lambda C(s,a) - \nu)/\alpha)\,] - \lambda\beta - \nu$$

Wasserstein distance:

$$D(d, d^R) = \sup_{\|g\|_L\leq 1} \mathop{\mathbb{E}}_{(s,a)\sim d}[g(s,a)] - \mathop{\mathbb{E}}_{(s,a)\sim d^R}[g(s,a)]$$

Wasserstein objective:

$$\max_{\substack{\lambda\geq 0,\nu,\\ Q_c:\mathcal{A}\times\mathcal{A}\to\mathbb{Z},\|g\|_L\leq 1}} -\mathop{\mathbb{E}}_{(s,a)\sim d^R}[g(s,a)] - \lambda\beta - \nu$$
$$g(s,a) = P^\pi Q_c(s,a) - Q_c(s,a) - \lambda C(s,a) - \nu$$

Wasserstein Entropy distance:

$$D(d, d^R) = \sup_{\|g\|_L\leq 1} \mathop{\mathbb{E}}_{(s,a)\sim d}[g(s,a) + \log(d(s,a))] - \mathop{\mathbb{E}}_{(s,a)\sim d^R}[g(s,a)]$$

Wasserstein Entropy objective:

$$\max_{\substack{\lambda\geq 0,\nu,\\ Q_c:\mathcal{A}\times\mathcal{A}\to\mathbb{Z},\|g\|_L\leq 1}} -\sum_{(s,a)} \exp(x(s,a) - 1) - \mathop{\mathbb{E}}_{(s,a)\sim d^R}[g(s,a)] - \lambda\beta - \nu$$
$$x(s,a) \coloneqq P^\pi Q_c(s,a) - Q_c(s,a) - \lambda C(s,a) - \nu + g(s,a)$$

Constrained Offline Policy Optimization

Algorithm

The optimization routine given by the constrained projection can be used within the COPO algorithm to obtain a safe policy with visitation density nearest to the reward optimal.

The constrained projection optimization is denoted by COPO in the algorithm sketch.

If the data was collected by a reward optimal policy, just run constrained projection, otherwise obtain the reward optimal policy and visitation density.

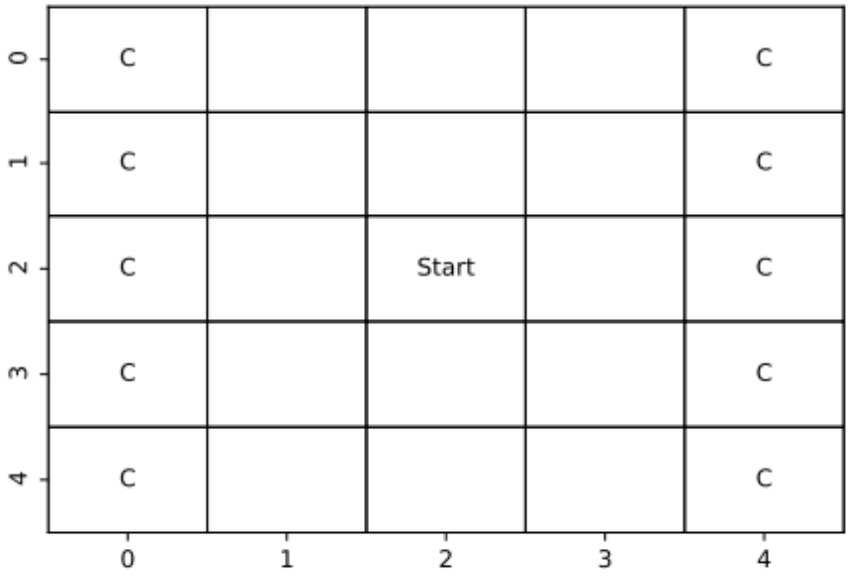
This can be done with offline policy optimization algorithms like AlgaeDICE [5], and offline distribution correction estimation algorithms like DualDICE [6].

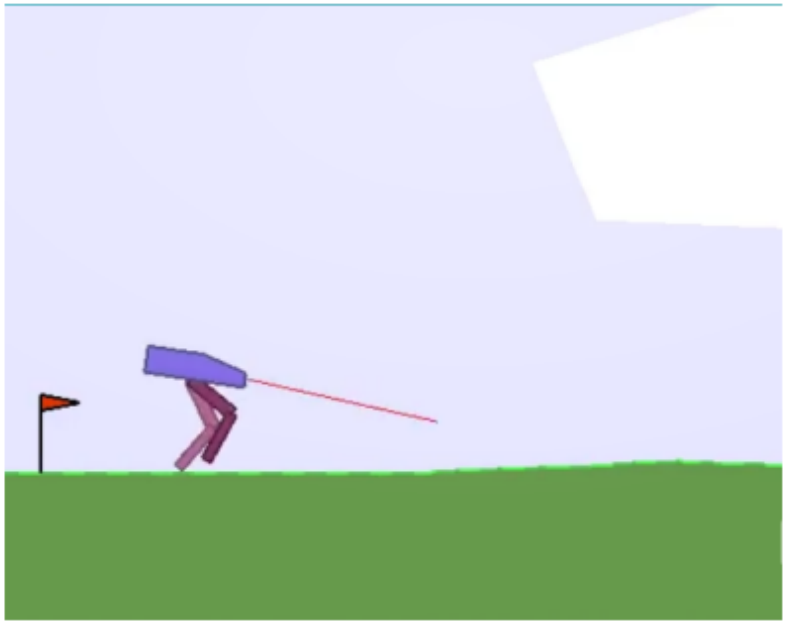
Algorithm 1 COPO Algorithm Sketch

Input Dataset $\mathcal{D} = \{s_i, a_i, r_i, s_i'\}_{i=0}^{n}$
Offline policy optimization algorithm, $\mathcal{A}$
Offline DICE algorithm, $\mathcal{P}$

if $\mathcal{D}$ is collected by a reward optimal policy **then**
  $\pi_C \leftarrow \text{COPO}(\mathcal{D})$
else
  Approximate reward optimal policy π_R by running $\mathcal{A}(\mathcal{D})$
  Approximate reward optimal policy visitation density d^{π_R} by running $\mathcal{P}(\mathcal{D}, \pi_R)$
  $\pi_C \leftarrow \text{COPO}(d^{\pi_R})$
end if
return π_C

Experiments





We conducted experiments in two environments: a custom gridworld with costs (left), and a robotic manipulation environment (right). In the gridworld, rewards are given for walking around the grid, further away from the center yields more reward. Costs are provided for walking on the left and right edges.

For the bipedal walker, reward is given for walking towards the right of the screen and costs are provided for exceeding a linear velocity constraint.

For gridworld, datasets were collected using random policies. For the Bipedal Walker environment, reward optimal policies were obtained using AlgaeDICE, and subsequently used for data collection.

Results

We compare COPO to the Batch Policy Learning under Constraints (BPLC) [7] method.

We employ the Wasserstein with entropy term distance in our experiments.

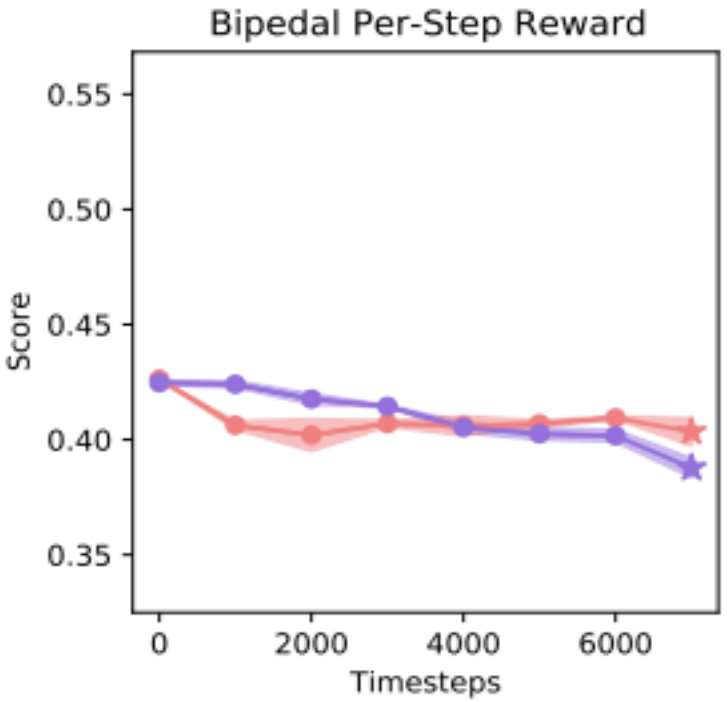
Each algorithm receives the same data set.

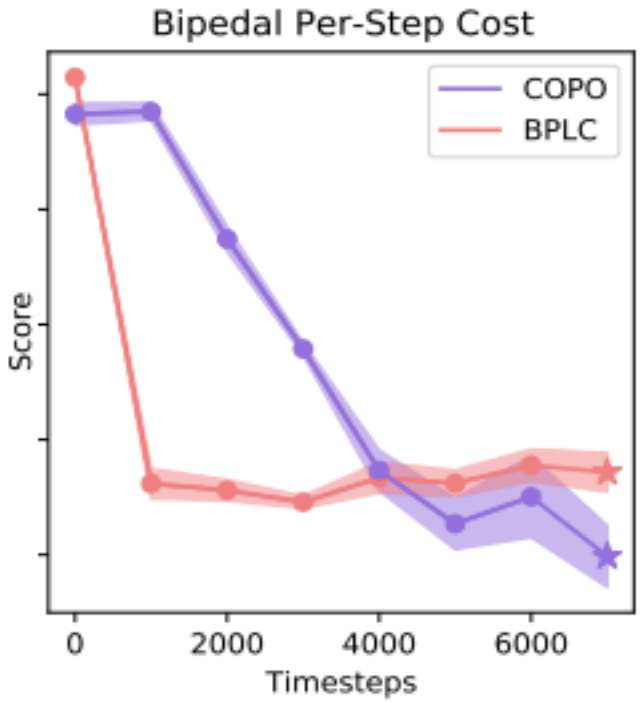
Each experiment was conducted over 10 random seeds.

Error bars are provided on the graphs, cost spread in the table represents the cost deviation.

The budget for the Bipedal Walker environment was set to 0.35

	Cost constraint satisfied		Cost spread (lower is better)		Mean per-step reward (higher is better)	
Budget	COPO	BPLC	**COPO**	BPLC	**COPO**	BPLC
0.4	Yes	Yes	$\mathbf{7.396e^{-3}}$	$3.714e^{-2}$	**0.761**	0.742
0.5	Yes	Yes	$\mathbf{8.253e^{-3}}$	$4.518e^{-2}$	**0.840**	0.779
0.6	Yes	Yes	$6.990e^{-2}$	$\mathbf{4.001e^{-2}}$	**0.901**	0.812
0.7	Yes	Yes	$\mathbf{9.921e^{-4}}$	$3.276e^{-2}$	**0.992**	0.866
0.8	Yes	Yes	$\mathbf{9.856e^{-4}}$	$3.230e^{-2}$	**0.992**	0.901
Average	—	—	$\mathbf{\mathit{1.750e^{-2}}}$	$\mathit{3.750e^{-2}}$	***0.897***	*0.821*





Contributions

Constrained DICE – Our constrained projection optimization routine is the first application of DICE estimation in the constrained RL setting. Using DICE estimation to mitigate effects of distributional shift in safety-critical RL problems is of practical importance for many applications.

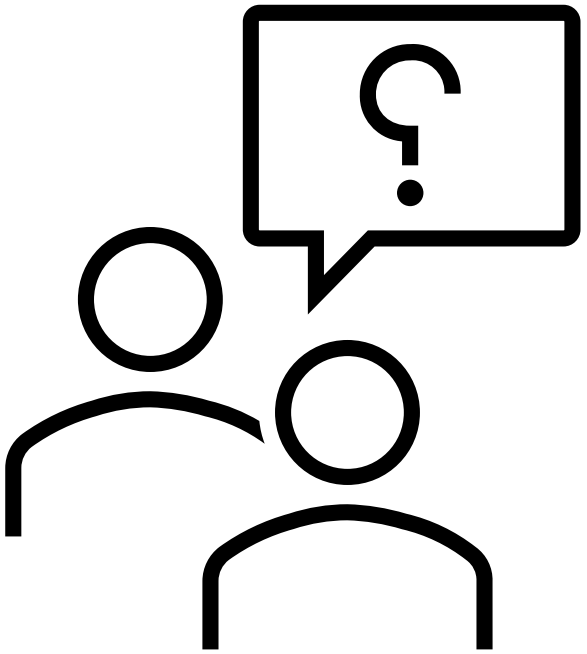
COPO Algorithm – We have shown how to use the constrained projection in a practical algorithm, titled COPO. We have also demonstrated how COPO improves performance against the state-of-the-art in constrained tabular and continuous control environments.

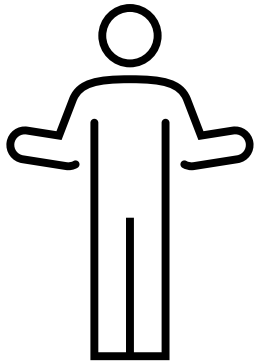
Confidence Intervals – We have extended the CoinDICE framework to our constrained projection functional, allowing us to obtain confidence intervals on the expected cost of the returned policy. These confidence intervals answer the question of how would the cost of the returned policy change had we seen a different data distribution. Details are in the paper.

What Next – Addressing theoretical limitations, characterizing the impact of the distance function on empirical performance, and demonstration in physical, non-simulated environments.

References

[1] Levine, S., Kumar, A., Tucker, G., and Fu, J. Offline reinforcement learning: Tutorial, review, and perspectives on open problems, 2020.
[2] Altman, E. Constrained markov decision processes. Chapman and Hall/CRC, 1999.
[3] Nachum, O. and Dai, B. Reinforcement learning via Fenchel-Rockafellar duality, 2020.
[4] Boyd, S. and Vandenberghe, L. Convex Optimization. Cambridge University Press, 2004.
[5] Nachum, O., Dai, B., Kostrikov, I., Chow, Y., Li, L., and Schuurmans, D. Algaedice: Policy gradient from arbitrary experience, 2019b
[6] Nachum, O., Chow, Y., Dai, B., and Li, L. Dualdice: Behavior-agnostic estimation of discounted stationary distribution corrections. Advances in Neural Information Processing Systems, volume 32. Curran Associates, Inc., 2019a
[7] Le, H., Voloshin, C., and Yue, Y. Batch policy learning under constraints. In Chaudhuri, K. and Salakhutdinov, R. (eds.), Proceedings of the 36th International Conference on Machine Learning, volume 97 of Proceedings of Machine Learning Research, pp. 3703–3712. PMLR, 09–15 Jun 2019.
[8] Dai, B., Nachum, O., Chow, Y., Li, L., Szepesvari, C., and Schuurmans, D. Coindice: Off-policy confidence interval estimation. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), Advances in Neural Information Processing Systems, volume 33, pp. 9398–9411. Curran Associates, Inc., 2020.





THANK YOU!
Questions?