Minimally Subdominant Decision Transformer

Rushit N. Shah, Brian D. Ziebart

Department of Computer Science
University of Illinois Chicago
Chicago, IL 60607
rshah231@uic.edu, bziebart@uic.edu

Abstract

A lesser-investigated challenge to effective offline reinforcement learning is the assumed access to a *scalar* reward function, which is not only hard to engineer, but often insufficient for capturing diverse user preferences as a single metric. To mitigate this we combine ideas from multiobjective optimization to develop an offline RL approach that allows an agent to learn behavior via multiple, interpretable user-specified metrics, rather than a hand-engineered scalar reward. Subdominance is a margin-based, hinge loss that measures the quality of one behavior trajectory relative to another based on an arbitrary number of cost metrics; minimizing subdominance explicitly encourages better-than-demonstrated behavior on the specified cost metrics. Recently-proposed autoregressive decision transformer models perform offline RL via history-conditioned, next-token prediction; while powerful, these methods self-report performance degradation with increasing suboptimality in the training data. By employing negative subdominance loss as a surrogate for the true reward signal we show that our method, **Min**imally **Sub**dominant **D**ecision **T**ransformer (**MinSubDT**), can consistently learn better-than-demonstrated behavior even when the demonstrated data is suboptimal. We also demonstrate how subdominance-minimizing cost feature representations can be learned from offline demonstration data. Our experiments are performed on Mujoco robotics tasks with offline RL benchmark datasets.

Introduction

In this paper, we combine powerful policy architectures (from offline RL literature) and imitation loss functions (from online, better-than-demonstrator IRL literature) to develop an approach that aims to explicitly learn learn better-than-demonstrator behavior policies offline from suboptimal demonstration data using a general loss function that integrates *any* user-specified cost metrics.

Preliminaries and Related Work

Offline Reinforcement Learning

We define a Markov decision process (MDP), $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, \mathcal{R})$ characterized by states $s \in \mathcal{S}$, actions $a \in \mathcal{A}$, transition dynamics $P(s'|s,a)$, and reward function $r = \mathcal{R}(s,a)$. A policy $\pi : \mathcal{S} \to \mathcal{A}$ maps a state to a specific action. A trajectory $\xi = (s_0, a_0, r_0, \ldots, s_T, a_T, r_T)$ is a sequence of states s_t, the actions a_t, and the corresponding reward received r_t, for $t = 0, \ldots, T$. The return of a trajectory at timestep t is the sum of future rewards starting from t, and is computed as $R_t = \sum_{t'=t}^{T}$. In reinforcement learning an optimal policy is sought that maximizes the expected return from $t = 1$, $\pi^* = \arg\max_\pi \mathbb{E}_\pi \left[\sum_{t=1}^{T} r_t\right]$. The core formulation of offline RL is identical to traditional online reinforcement learning, with the key difference being the training data is collected via some (unknown) behavior policy π_β, and the agent's policy π is not allowed to interact with the environment.

Transformers and Offline Sequence Modeling

Decision Transformer (DT) (Chen et al. 2021) uses a causally-masked transformer (Vaswani et al. 2017; Radford et al. 2018) as policy architecture to predict actions that maximize returns conditioned on the observation history and a target return. Their architecture consists of stacks of self-attention layers, which enable learning associations between each previous "token" (or observation) in the sequence. The trajectory representation developed therein is key to enabling the use of transformers; a trajectory is represented as a sequence of states, actions, and *returns* (not rewards) as $\xi = (R_1, s_1, a_1, R_2, s_2, a_2, \ldots, R_T, s_T, a_T)$, where $R_t = \sum_{t'=t}^{T} r_t$ is the future return.We note that methods discussed thus far assume access to the true reward function of the environment.

Subdominance Minimization

Given a set of trajectory-level cost metrics $\mathbf{f} : \Xi \to \mathbb{R}^K_{\geq 0}$, the margin-based, absolute subdominance (Ziebart et al. 2022) of trajectory ξ relative to another trajectory $\tilde{\xi}$ is defined for each feature f_k as:

$$\text{subdom}^k_{\alpha_k,\beta_k}(\xi, \tilde{\xi}) \triangleq \left[\alpha_k(f_k(\xi) - f_k(\tilde{\xi})) + \beta_k\right]_+ \quad (1)$$

where $[x]_+ \triangleq \max(x, 0)$ denotes the hinge function, β_k is the required margin by which ξ is required to dominate $\tilde{\xi}$ in feature k, and α_k is the relative sensitivity of feature k. The individual feature subdominance measures

AAAI 2024 Spring Symposium on User-Aligned Assessment of Adaptive AI Systems. Stanford University, Stanford, CA, USA.

Dataset	Environment	MinSubDT (Ours) Engg. Cost Metrics	MinSubDT (Ours) Learned Cost Metrics	Offline RL DT	CQL	BEAR	BRAC-v	AWR	BC
Medium-Expert	HalfCheetah	**89.5**	86.8	87	62.4	53.4	41.9	52.7	59.9
Medium-Expert	Hopper	110.7	**110.8**	107.6	**111.0**	96.3	0.8	27.1	79.6
Medium-Expert	Walker	108.1	**108.5**	108.1	98.7	40.1	81.6	53.8	36.6
Medium	HalfCheetah	41.3	40.9	42.6	44.4	41.7	**46.3**	37.4	43.1
Medium	Hopper	56.8	55.8	**67.6**	58.0	52.1	31.1	35.9	63.9
Medium	Walker	70.4	**74.9**	74.0	79.2	59.1	**81.1**	17.4	77.3
Medium-Replay	HalfCheetah	32.3	31.8	36.6	46.2	38.6	**47.7**	40.3	4.3
Medium-Replay	Hopper	40.5	37.1	**82.7**	48.6	33.7	0.6	28.4	27.6
Medium-Replay	Walker	37.1	36.0	**66.6**	26.7	19.2	0.9	15.5	36.9
Average		65.2	64.7	**74.7**	63.9	48.2	36.9	34.3	46.4
Average (w/o Medium-Replay)		79.5	79.6	**81.1**	75.6	57.1	47.1	37.4	60.1

Table 1: Expert-normalized scores for the D4RL datasets. We report the mean and variance for three seeds.

for all $k = 1, \ldots, K$ from equation 1 are then sum-aggregated into total subdominance as $\text{subdom}^{\Sigma}_{\boldsymbol{\alpha},\boldsymbol{\beta}}(\xi, \tilde{\xi}) \triangleq \sum_k \text{subdom}^{k}_{\alpha_k,\beta_k}(\xi, \tilde{\xi})$. [1]

MinSubDT (Our Formulation)

We begin by defining the minimum subdominance of a trajectory ξ with respect to a set of trajectories $\tilde{\Xi} = \{\tilde{\xi}_1, \tilde{\xi}_2, \ldots, \tilde{\xi}_N\}$ as

$$\text{subdom}^{[\Sigma]}_{\boldsymbol{\alpha}^*,\boldsymbol{\beta}^*}(\xi, \tilde{\Xi}) = \min_{\boldsymbol{\alpha},\boldsymbol{\beta} \geq 0} \frac{1}{|\tilde{\Xi}|} \sum_{\tilde{\xi} \in \tilde{\Xi}} \text{subdom}^{[\Sigma]}_{\boldsymbol{\alpha},\boldsymbol{\beta}}(\xi, \tilde{\xi}). \quad (2)$$

The minimum subdominance (Equation 2) is a (cost) estimate of the performance of trajectory ξ relative to the demonstrated trajectory set $\tilde{\Xi}$. We can then use the *negative* subdominance to compute a surrogate reward signal r'_t at each timestep t in trajectory ξ as: $r'_t = 0$ if $t < T$, and $r'_t = -\text{subdom}^{[\Sigma]}_{\boldsymbol{\alpha}^*,\boldsymbol{\beta}^*}(\xi, \tilde{\Xi})$ if $t = T$. Computing the surrogate reward at the final timestep T avoids having to determine the correct credit assignment for each state $t \neq T$ in the trajectory. Finally, a trajectory's surrogate returns R'_t starting from t can be computed as $R'_t = \sum_{t'=t}^{T} r'_{t'}$. Then, leveraging the transformer-compatible trajectory representation of Chen et al. (2021), we represent each trajectory $\tilde{\xi}' \in \tilde{\Xi}'$ as $\tilde{\xi}' = (R'_1, s_1, a_1, R'_2, s_2, a_2, \ldots, R'_T, s_T, a_T)$ where the surrogate return R'_t is computed as defined previously. With this surrogate trajectory representation available, the autoregressive architecture can be applied exactly as in Decision Transformer to predict actions that maximize the future negative subdominance (same as minimizing the subdominance), rather than maximizing the true returns, thereby decoupling our approach from relying on the true, underlying reward/cost function.

Learning Cost Representation A set of trajectory cost metrics $\mathbf{f} : \Xi \to \mathbb{R}^K_{\geq 0}$ is required to compute subdominance (Ziebart et al. 2022). In essence, these metrics are properties of an environment that, when minimized over a trajectory, allow an agent to successfully complete a task; hand-engineering these can still be a significant burden in many domains. To mitigate this, we propose to *l*earn a set of cost metrics $\mathbf{f}_\psi$ from pairwise preferences over demonstrations. Given pairwise preferences over demonstrated trajectories $\tilde{\mathcal{D}} = \{\tilde{\xi}_i \prec \tilde{\xi}_j | \tilde{\xi}_i, \tilde{\xi}_j \in \tilde{\Xi}\}$, a preference-preserving (latent) representation $\mathbf{f}_\psi : \mathcal{S} \to \mathbb{R}^{K'}_{\geq 0}$ (of dimensionality K') can be learned by minimizing subdominance via

$$\arg\min_{\mathbf{f}_\psi \in \mathcal{F}} \mathbb{E}_{(\tilde{\xi}_i \prec \tilde{\xi}_j) \sim \tilde{\mathcal{D}}} \left[-\log \frac{e^{-c_{j,i}}}{e^{-c_{i,j}} + e^{-c_{j,i}}} \right], \quad (3)$$

where cost $c_{i,j} = \text{subdom}_{\alpha,\beta}(\mathbf{f}_\psi(\tilde{\xi}_i), \mathbf{f}_\psi(\tilde{\xi}_j))$, and $x \prec y$ indicates a preference for y over x. The cost representation $\mathbf{f}_\psi$ thus learned can be employed directly to compute subdominance, and subsequently surrogate rewards. This formulation is most similar to Trajectory-Ranked Reward Extrapolation (T-REX) (Brown et al. 2019) but differs in one key aspect: our formulation permits learning latent representations of *any* dimensionality, rather than just a scalar cost signal; this allows us to recover multiple, competing objectives from preferences, rather than arbitrarily extrapolating over a scalar reward signal.

Experiments

For our experiments, we choose three continuous control, Mujoco robotics tasks: HalfCheetah, Hopper, and Walker. The engineered cost metrics comprise the x-velocity (for HalfCheetah), and x and z velocities (for Hopper and Walker) monotonically transformed into cost metrics; doing so allows the agent to maximize these velocities and thereby complete the respective tasks. For reliable comparison, we follow the same training methodology as for Decision Transformer (Chen et al. 2021), use the same D4RL benchmark data (Fu et al. 2020) and compare against the baselines therein.

Results The results in Table 1 are normalized scores (% relative to unnormalized score of an online SAC agent trained using the true reward function). MinSubDT outperforms DT and other baselines on 4 out of the 9 tasks. Overall, MinSubDT (using engineered and learned cost metrics) without access to the exact underlying reward has an average performance within 0.5 percentage points of DT.

[1] Intuitively, minimizing subdominance with respect to a set of demonstrated trajectories explicitly encourages an agent to outperform those trajectories i.e., incur lower cost features, by a margin β_k in each cost feature dimension f_k.

References

Brown, D. S.; Goo, W.; Nagarajan, P.; and Niekum, S. 2019. Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. *arXiv preprint arXiv:1904.06387*.

Chen, L.; Lu, K.; Rajeswaran, A.; Lee, K.; Grover, A.; Laskin, M.; Abbeel, P.; Srinivas, A.; and Mordatch, I. 2021. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34: 15084–15097.

Fu, J.; Kumar, A.; Nachum, O.; Tucker, G.; and Levine, S. 2020. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*.

Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I.; et al. 2018. Improving language understanding by generative pre-training.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Ziebart, B.; Choudhury, S.; Yan, X.; and Vernaza, P. 2022. Towards Uniformly Superhuman Autonomy via Subdominance Minimization. In Chaudhuri, K.; Jegelka, S.; Song, L.; Szepesvari, C.; Niu, G.; and Sabato, S., eds., *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, 27654–27670. PMLR.