In-Context Multi-Armed Bandits via Supervised Pretraining

Fred Weiying Zhang
Department of Statistics and Data Science
Yale University
New Haven, CT 06511
fred.zhang@yale.edu

Jiaxin Ye
Department of Mathematcis
University of California, San Diego
San Diego, CA 92122
jiye@ucsd.edu

Zhuoran Yang
Department of Statistics and Data Science
Yale University
New Haven, CT 06511
zhuoran.yang@yale.edu

Abstract

Exploring the in-context learning capabilities of large transformer models, this research focuses on decision-making within reinforcement learning (RL) environments, specifically multi-armed bandit problems. We introduce the Reward Weighted Decision-Pretrained Transformer (DPT-RW), a model that uses straightforward supervised pretraining with a reward-weighted imitation learning loss. The DPT-RW predicts optimal actions by evaluating a query state and an in-context dataset across varied tasks. Surprisingly, this simple approach produces a model capable of solving a wide range of RL problems in-context, demonstrating online exploration and offline conservatism without specific training in these areas. A standout observation is the optimal performance of the model in the online setting, despite being trained on data generated from suboptimal policies and not having access to optimal data.

1 Introduction

The multi-armed bandit problem, a canonical challenge in reinforcement learning, is often used to study the trade-offs between exploration and exploitation. The metric of regret serves as a crucial benchmark to gauge the performance of algorithms in this context. Our algorithm, reward-weighted decision-pretrained transformer (DPT-RW) contribute notably to this body of work, and the primary highlights are as follows:

- Near-optimal, logarithmic regret: Our algorithm achieves near-optimal regret and matches the performance of the algorithm that has access to the optimal policy during pretraining.
- Online exploration from offline data training: What makes our approach particularly noteworthy is its training exclusively on offline data using a reward-reweighted imitation learning loss. Our results come in spite of the fact that many of the offline observations are generated through highly suboptimal policies and the lack of explicit exploration instruction.

In summary, our research offers a novel perspective on the multi-armed bandit problem by achieving near-optimal performance solely through offline data and without explicit exploration programming or optimal policy sampling.

2nd Foundation Models for Decision Making Workshop at NeurIPS 2023.

Comparison with Lee et al. Lee et al.'s "Supervised Pretraining Can Learn In-Context Reinforcement Learning" [7] offers a compelling exploration into supervised pretraining's potential in in-context reinforcement learning. Their methodology, highlighting the benefits of supervised pretraining for reinforcement learning models to effectively adapt using offline samples, has been an influential precursor to our own research direction.

A salient assumption in their approach is the capability to draw samples from the optimal policy, which they use to compute a negative log-likelihood loss. While this serves as a foundational pillar for their theoretical results, its real-world applicability is often constrained due to the complexities associated with securing or approximating optimal policies across varied environments.

Diverging from this paradigm, our methodology forgoes the need to sample from the optimal policy. We introduce an imitation learning loss, a critical component considering the inherent challenge in reinforcement learning: an agent is unaware of the resulting state and reward for actions it hasn't performed. Paired with this, our reward reweighting is essential. It ensures that agents are properly incentivized towards behaviors that accrue high rewards, making it a central mechanism to channel the agent's learning trajectory in the desired direction.

Together, these elements render our approach a more practical algorithm for real-world offline datasets with minimal loss in performance.

2 Related Works

In-context learning The paradigm of in-context learning has emerged as a transformative approach, emphasizing the ability of models to generalize from limited examples by extracting knowledge from a provided context [9]. This principle is shared with few-shot and zero-shot learning but is distinctively highlighted in large-scale transformer models, which can adapt and generalize over various tasks when given suitable contextual prompts. Brown et al. [2] demonstrated this ability, but it's worth noting that Garg et al. delved deeper into what transformers can learn in-context by examining simple function classes [4]. They provided insights into the limitations and strengths of transformers in capturing different functional patterns. Bai et al. showcased that transformers can be seen as statisticians, presenting a framework for provable in-context learning combined with in-context algorithm selection [1]. Their work suggests that not only can transformers learn in-context, but they can also be harnessed to make algorithmic decisions based on the context.

Foundation models for decision making Foundation models, which are pre-trained on vast data and fine-tuned for specialized tasks, have exhibited an increasingly significant role in the AI landscape. These models capitalize on a broad foundational knowledge, enabling the building of specific capabilities, thereby leading to training economies of scale [13]. For decision-making, such models have been instrumental, with models like the decision transformer based off of the GPT-series demonstrating aptitude in decision-making scenarios [3].

Offline reinforcement learning Traditional RL methods engage in active interactions with environments to formulate policies. However, real-world applications often present scenarios where such active engagements are either too risky or unfeasible. Offline RL addresses this by gleaning insights from a static dataset devoid of further environment interactions [8]. The power of offline RL is in harnessing ample pre-collected data to derive impactful policies. In this regard, methods such as Conservative Q-Learning (CQL) [6] and Bootstrapping Error Accumulation Reduction (BEAR) [5] have pushed the frontier, showing robust performance, especially when online data collection is limited or prohibitively costly.

Reward weighting The weighting of rewards in Reinforcement Learning is a nuanced approach that alters the dynamics of learning, promoting the behavior of the agent with high reward. One notable contribution in this domain is the work by Peters et al. [10] Their work provided a regression-based approach for operational space control, emphasizing the potency of reward-weighted regression techniques in facilitating smooth policy search. Building off of this, Peng et al. introduced Advantage-Weighted Regression, offering a novel perspective by leveraging advantage estimations to weight regression updates [11]. These methods bridge the domains of reward-weighting and off-policy learning, presenting scalable solutions to RL challenges, especially in environments with complex reward dynamics.

3 Algorithm

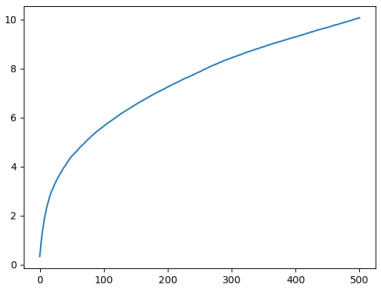
The basic structure of our algorithm closely follows the work of Lee et al. [7]. Logically, we keep their pretraining and testing specifications for consistency.

Algorithm 1 Decision-Pretrained Transformer Reward Weighted (DPT-RW): Training and Deployment

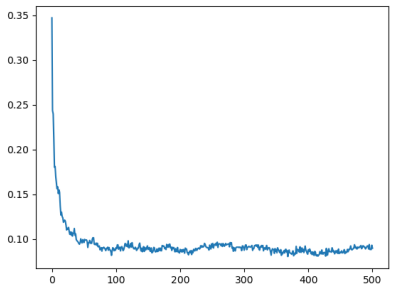
// Collecting pretraining dataset
Initialize empty pretraining dataset $\mathcal{B}$
for i in $[N]$ **do**
    Sample task $\tau \sim \mathcal{T}_{pre}$, in-context dataset $D \sim \mathcal{D}_{pre}(\cdot;\tau)$, query state $s_{query} \sim D_{query}$
    Sample label $a \sim P_a$ and add (s_{query}, D, a) to $\mathcal{B}$
end for
// Pretraining model on dataset
Initialize model M_θ with parameters θ
for i in $[E]$ **do**
    Sample (s_{query}, D, a) from $\mathcal{B}$ and predict $\hat{p}_j(\cdot) = M_\theta(\cdot|s_{query}, D_j)$ for all $j \in [n]$
    Compute loss in eq:pretrain-obj with respect to a and backpropagate to update θ.
end for
// Offline test-time deployment
Sample unknown task $\tau \sim \mathcal{T}_{test}$, sample dataset $D \sim \mathcal{D}_{test}(\cdot;\tau)$
Deploy M_θ in τ by choosing $a_h \in_{a\in\mathcal{A}} M_\theta(a|s_h, D)$ at step h
// Online test-time deployment
Sample unknown task $\tau \sim \mathcal{D}_{test}$ and initialize empty $D = \{\}$
for `ep` in `max_eps` **do**
    Deploy M_θ by sampling $a_h \sim M_\theta(\cdot|s_h, D)$ at step h
    Add $(s_1, a_1, r_1, \ldots)$ to D
end for

Basic decision models. In the context of our study, we narrow our focus to the multi-armed bandit problem, which is a special case of the Markov decision process (MDP). A multi-armed bandit problem can be specified by a tuple $\zeta = \langle \mathcal{A}, R \rangle$, where $\mathcal{A}$ is the action space and $R : \mathcal{A} \to \Delta(R)$ is the reward function. Unlike the general MDP, the state space is trivial (a single state) and there is no state transition, which simplifies the model to: (1) the learner selects an action a from $\mathcal{A}$; (2) a reward r is received according to $R(\cdot|a)$. This process can be repeated for a specified number of trials or indefinitely. A policy π in this context maps from the single state to a probability distribution over actions and is used to determine which arm to pull in each round. We denote the optimal policy as π^* which maximizes the expected total reward $V(\pi^*) = \max_\pi V(\pi) := \max_\pi \mathbb{E}_\pi \left[\sum_h r_h\right]$. The multi-armed bandit problem requires strategically deciding which arm to pull (i.e., which action to take) in each round in order to maximize cumulative reward, often under the constraint of limited knowledge about the true reward distributions of each arm.

Pretraining. In establishing the pretraining task distribution, denoted $\mathcal{T}_{pre}$, we generate 5-armed bandits, represented as $|A| = 5$. The associated reward function for a particular arm a adheres to a normal distribution, formalized as $R(\cdot|s, a) = N(\mu_a, \sigma^2)$, where μ_a is independently sampled from a uniform distribution Unif$[0, 1]$ and σ is fixed at 0.3. For the creation of in-context datasets, designated D_{pre}, we employ a strategy of random action frequency generation. This involves sampling probabilities using a Dirichlet distribution, which are then combined with a point-mass distribution on a single, randomly-selected arm, together forming the action distribution P_a. Subsequent action sampling adheres to this particular distribution, ensuring that the data-generating policies have adequate coverage of the probability simplex. The optimal policy π^*_τ for a given bandit τ is determined by ${}_a\mu_a$. Model M_θ is pretrained to predict a^* from D and is applicable for datasets with a size up to $n = 500$, i.e. we want to train a transformer $M_\theta(\cdot|\cdot)$ s.t. $M_\theta(a_n|s_1, a_1, ..., s_{n-1}, a_{n-1}, s_n) \approx \pi^*(a_n|s_n)$ for each task where π^* is the optimal policy, and $(s_1, a_1, ..., s_n)$ are generated by M_θ. Formally, we aim to train a causal GPT-2 transformer model M parameterized by θ, which outputs a distribution over actions $\mathcal{A}$, to minimize the expected loss over samples from the pretraining distribution: $\min_\theta \mathbb{E}_{P_{pre}} \sum_{j\in[n]} \ell\left(M_\theta(\cdot \mid s_{query}, D_j), \hat{a}\right)$ Specifically, we set the loss to be the weighted cross



(a) label 1



(b) label 2

Figure 1: (a) Offline performance on in-distribution bandits, given random in-context datasets. (b) Online cumulative regret of the same model. The mean and standard error are computed over 500 test tasks.

entropy for each $j \in [n]$: $-\sum_{j\in[n]} \max(\bar{r}_{a_j} - \bar{r}_j, 0) \cdot \log M_\theta(a_j \mid s_{query}, D_j)$ where $\bar{r}$ is the average reward over the entire trajectory, and $\bar{r}_a$ is the average reward of arm a sample over the entire trajectory. Note that there is no weighting in the vanilla DPT and a_j is replaced with a^*.

Our choice of loss function is motivated by reward-weighted regression [10], where we incentivize the model to select actions with greater reward. We subtract the average reward across each trajectory as a normalization. Another key ingredient in our pre-training process is the use of the early stopping hyperparameter E, which serves as a regularization.

4 Results

Initiating with a practical analysis of DPT-RW within the context of a multi-armed bandit, which is a comprehensively examined specific instance of the MDP where there is a singular state space S and a single-step horizon $H = 1$, the performance of DPT-RW will be scrutinized. This will be done in contexts of choosing a favorable action based on historical data from offline scenarios, as well as during online learning, where the objective revolves around optimizing the cumulative reward starting from the empty dataset. In an offline scenario, it becomes imperative to consider uncertainty emanating from noise since certain actions might be inadequately sampled. Conversely, in an online scenario, the essentiality lies in astutely maintaining a balance between exploration and exploitation to minimize the overarching regret.

The metric we use to evaluate the algorithm in offline setting is the suboptimality $\mu_{a^*} - \mu_{\hat{a}}$ where $\hat{a}$ is the chosen action, while the metric we use to evaluate the algorithm in online setting is the cumulative regret $\sum_k \mu_{a^*} - \mu_{\hat{a}_k}$ where $\hat{a}_k$ is the kth chosen action.

DPT-RW exhibits a notable proficiency in reasoning through uncertainty. In the offline scenario, as depicted in Figure 1a, when in-context datasets are derived from the same distribution as utilized during pretraining, DPT-RW markedly excels in its performance. Achieving a performance metric of 10^{-1} after 50 trials. The outcomes suggest that the transformer possesses the capacity to work through uncertainty, especially that which is brought about by the noisy rewards within the dataset.

In a compelling turn, employing the same transformer but opting to sample actions rather than utilizing an argmax produces a particularly effective online bandit algorithm, as illustrated in Figure 1b. The cumulative regret exhibits a logarithmic trend, with final regret just above 10. This is exactly the same as the performance of the vanilla DPT and the known optimal algorithm UCB. Even though DPT-RW was trained entirely throuh weighted imitation learning, it still achieves optimal regret in the bandit setting.

5 Discussion

An initial foray into the intersection of in-context learning and offline reinforcement learning was achieved by the work of Lee et al. and their focus on supervised pretraining. However, an essential step of their method is the reliance on sampling from the optimal policy. While providing insightful theoretical results, this assumption poses challenges in real-world scenarios where access to such sampling is often impractical.

Adapting to these practical constraints, our algorithm uses an imitation learning loss combined with reward reweighting. This subtle yet impactful adjustment not only mitigates the need for sampling from the optimal policy but also broadens the realm of tasks the algorithm can effectively handle.

Moreover, the efficacy of our approach isn't solely attributed to the imitation learning loss and reward reweighting. An additional layer of finesse comes into play through the tuning of an early stopping parameter. This tuning is pivotal in minimizing the online regret, ensuring that we do not overfit to the online regret.

The practical implications of these modifications become evident in the multi-armed bandit scenario. Despite having access to strictly less information, our algorithm delivers performance equal with theirs. However, as the focus shifts to environments characterized by continuous state and action spaces, the necessity for a value network becomes pronounced.

Of particular intrigue is the observation that our algorithm showcases strong online performance, specifically midway between the initialization and convergence of the imitation learning loss. This phenomenon prompts deeper exploration into the inherent learning dynamics. Instead of mere methodology refinement, we're invested in uncovering the underlying reasons for this behavior.

As part of our ongoing endeavors, linear bandits and Markov Decision Processes (MDPs) are areas we have begun to investigated. We anticipate that insights derived from these domains can shed light on the observed learning dynamics.

References

[1] Bai, Y., Chen, F., Wang, H., Xiong, C., Mei, S. Transformers as Statisticians: Provable In-Context Learning with In-Context Algorithm Selection, *37th Conference on Neural Information Processing Systems*, 2023.

[2] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, G., et al. Language models are few-shot learners, *34th Conference on Neural Information Processing Systems*, 2020.

[3] Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., Abbeel, P., Srinivas, A., Mordatch, I. Decision transformer: Reinforcement learning via sequence modeling, *35th Conference on Neural Information Processing Systems*, 2021.

[4] Garg, S., Tsipras, D., Liang, P.S., Valiant, G. What can transformers learn in-context? a case study of simple function classes, *36th Conference on Neural Information Processing Systems*, 2022.

[5] Kumar, A., Fu, J., Tucker, G., Levine S. Stabilizing Off-Policy Q-Learning via Bootstrapping Error Reduction, *33rd Conference on Neural Information Processing Systems*, 2019.

[6] Kumar, A., Zhou, A., Tucker, G., Levine, S. Conservative Q-Learning for Offline Reinforcement Learning, *34th Conference on Neural Information Processing Systems*, 2020.

[7] Lee, J. N., Xie, A., Pacchiano, A., Chandak, Y., Finn, C., Nachum, O., Brunskill, E. "Supervised Pretraining Can Learn In-Context Reinforcement Learning", arXiv preprint arXiv:2306.14892, 2023.

[8] Levine, S., Kumar, A., Tucker, G., Fu, J. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. arXiv preprint arXiv:2005.01643, 2020.

[9] Min, S., Lyu, X., Holtzman, A., Artetxe, M., Lewis, M., Hajishirzi, H., Zettlemoyer, L. "Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?", *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing* 11048 - 11064.

[10] Peters, J., Schaal, S. Reinforcement Learning by Reward-weighted Regression for Operational Space Control, *24th Internationa Coneference on Machine Learning*, 2007.

[11] Peng, X. B., Kumar, A., Zhang, G., Levine, S. Advantage-Weighted Regression: Simple and Scalable Off-Policy Reinforcement Learning, arXiv preprint arXiv:1910.00177, 2023.

[12] Robbins, H. (1952) “Some aspects of the sequential design of experiments." In *Bulletin of the American Mathematical Society* **58**(5): 527–535.

[13] Yang, S., Nachum, O., Du, Y., Wei, J., Abbeel, P., Schuurmans, D.. Foundation models for decision making: Problems, methods, and opportunities. arXiv preprint arXiv:2303.04129, 2023.