Dueling Posterior Sampling for Preference-Based Reinforcement Learning

Ellen R. Novoseller [1] **Yanan Sui** [2] **Yisong Yue** [1] **Joel W. Burdick** [1]

Abstract

In preference-based reinforcement learning (PBRL), an agent interacts with the environment while receiving preferences instead of absolute feedback. While there is increasing research activity in PBRL, the design of formal frameworks that admit tractable theoretical analysis remains an open challenge. We present DUELING POSTERIOR SAMPLING (DPS), which employs preference-based posterior sampling to learn both the system dynamics and the underlying utility function that governs the user's preferences. To solve the credit assignment problem, we develop a Bayesian approach to translate user preferences to a posterior distribution over state/action reward models. We prove an asymptotic no-regret rate for DPS with Bayesian logistic regression credit assignment; to our knowledge, this is the first regret guarantee for PBRL. We also discuss possible avenues for extending this proof methodology to analyze other credit assignment models, and finally, evaluate the approach empirically.

1. Introduction

In many domains, ranging from clinical trials (Sui et al., 2018a) to autonomous driving (Sadigh et al., 2017) and human-robot interaction (Kupcsik et al., 2018), it can be unclear how to define a reward signal for reinforcement learning (RL). In such situations, the RL agent seeks to interact optimally with a human user; thus, rewards should reflect the extent to which the algorithm achieves the user's goals. Yet, for many systems, for instance in autonomous driving (Basu et al., 2017) and robotics (Argall et al., 2009; Akrour et al., 2012), users have difficulty with both specifying numerical reward functions and providing demonstrations of desired behavior. In such cases, the user's *preferences* form a more reliable measure of desired system behavior.

We thus study the problem of preference-based reinforcement learning (PBRL), where the RL agent executes a pair of trajectories, and the user provides (noisy) preference feedback regarding which trajectory has higher utility. While the study of PBRL has seen increased interest in recent years (Christiano et al., 2017; Wirth et al., 2017), it remains an open challenge to design formal frameworks that admit tractable theoretical analysis. Compared to the preference-based bandit setting, which has seen significant theoretical progress (Yue et al., 2012; Dudík et al., 2015; Wu & Liu, 2016; Sui et al., 2017; 2018b), one major challenge is how to address credit assignment when only receiving feedback at the trajectory level compared to the state/action level.

In this paper, we present DUELING POSTERIOR SAMPLING (DPS), which uses preference-based posterior sampling to tackle the PBRL problem in the Bayesian regime. Posterior sampling (also known as Thompson sampling) (Thompson, 1933; Osband et al., 2013; Gopalan & Mannor, 2015; Agrawal & Jia, 2017; Osband & Van Roy, 2017) is a Bayesian model-based approach to balancing exploration and exploitation, enabling the algorithm to efficiently learn models of both the environment's state transition dynamics and the reward function. Previous work on posterior sampling in RL (Osband et al., 2013; Gopalan & Mannor, 2015; Agrawal & Jia, 2017; Osband & Van Roy, 2017) all focused on learning from absolute rewards, while we show how to extend posterior sampling to both elicit and learn from trajectory-level preference feedback.

To elicit preference feedback, at every episode of learning, DPS draws two independent samples from the posterior to generate two trajectories. This approach is inspired by the Self-Sparring algorithm proposed for the bandit setting (Sui et al., 2017); however, our theoretical analysis is quite different, due to the need to incorporate trajectory-level preference learning and state transition dynamics.

To learn from preference feedback, DPS internally maintains a Bayesian state/action reward model that explains the preferences. This reward model is a solution to the *temporal credit assignment problem* (Akrour et al., 2012; Zoghi et al., 2014; Szörényi et al., 2015; Christiano et al., 2017; Wirth et al., 2016; 2017) and determines which of the encountered states and actions are responsible for the trajectory-level

[1] Department of Computing and Mathematical Sciences, California Institute of Technology, Pasadena, California, USA [2] Department of Computer Science, Stanford University, Stanford, California, USA. Correspondence to: Ellen R. Novoseller <enovoseller@caltech.edu>.

Real-world Sequential Decision Making workshop at ICML 2019.

preference feedback. Learning from trajectory preferences is in general a very challenging problem, as information about rewards is sparse, is only relative to the pair of trajectories being compared, and does not explicitly include information about actions within trajectories.

We developed DPS concurrently with an analysis framework for characterizing regret convergence in the episodic learning setting. We evaluate several possible Bayesian credit assignment models, and prove an asymptotic no-regret rate for DPS using Bayesian logistic regression (Albert & Chib, 1993; Murphy, 2012) as the credit assignment model. To our knowledge, this is the first PBRL approach with theoretical guarantees. In addition, we also demonstrate that DPS delivers competitive performance in simulation.

2. Problem Statement

Preliminaries. We consider fixed-horizon Markov Decision Processes (MDPs), in which rewards are replaced by preferences over trajectories. This class of MDPs can be represented as a tuple, $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \succ, \phi, p, p_0, h)$, where the state space $\mathcal{S}$ and action space $\mathcal{A}$ are finite sets. The agent, using policy π, episodically interacts with the environment with length-h roll-out trajectories of the form $\tau = \{s_0, a_0, s_1, a_1, \ldots, s_{h-1}, a_h, s_h\}$. Since we are eliciting preference feedback, in each episode i, the agent executes two roll-outs τ_{i1} and τ_{i2}, and observes a preference between the two. The initial state is sampled from p_0, while p defines the transition dynamics: $s_{t+1} \sim p(\cdot|s_t, a_t)$.

We use $\succ$ to denote the stochastic preference relationship between trajectories, and $\phi(\tau, \tau') = \mathbb{P}(\tau > \tau') - 0.5 \in [-0.5, 0.5]$ to capture the feedback generation mechanism. We assume that $\succ$ is a total ordering over trajectories, and $\tau \succ \tau' \Leftrightarrow \phi(\tau, \tau') > 0$. We use $\tau > \tau'$ to denote the event that trajectory τ was preferred over τ' in a preference elicitation, i.e., $\tau > \tau'$ is observed with probability $\phi(\tau, \tau') + 0.5$. We further assume an underlying utility function $\overline{r}(\tau)$ for each trajectory, such that $\tau \succ \tau' \Leftrightarrow \overline{r}(\tau) > \overline{r}(\tau')$, and define ϕ using $\overline{r}$. For instance, if the preferences are noiseless, then: $\phi(\tau_i, \tau_j) = \mathbb{I}[\overline{r}(\tau_i) > \overline{r}(\tau_j)] - 0.5$. We primarily assume a logistic or Bradley-Terry link function: $\phi_{\text{lin}}(\tau_i, \tau_j) := [1 + \exp(-c(\overline{r}(\tau_i) - \overline{r}(\tau_j)))]^{-1}$ with "temperature" $c \in (0, \infty)$. Our problem setting resembles the PSDP defined in (Wirth & Fürnkranz, 2013b), except that we also incorporate the noise model through which the underlying utilities are stochastically translated to preferences. Finally, we assume that the utilities decompose additively: $\overline{r}(\tau) \equiv \sum_{i=1}^{h} \overline{r}(s_i, a_i)$ for state/action pairs in τ.

Algorithm 1 Dueling Posterior Sampling (DPS)

```
$H = \emptyset$ {Initialize history}
$T = \emptyset$ {Initialize list of preference data}
Initialize prior for $f$ {Initialize state transition model}
Initialize prior for $g$ {Initialize utility model}
while True do
    $\pi_1 \leftarrow$ ADVANCE$(H, T, f, g)$
    $\pi_2 \leftarrow$ ADVANCE$(H, T, f, g)$
    Sample trajectories $\tau_1$ and $\tau_2$ from $\pi_1$ and $\pi_2$
    Observe feedback $b = \mathbb{I}(\tau_2 > \tau_1)$
    $H = H \cup (s_1^{\tau_1}, a_1^{\tau_1}, s_2^{\tau_1}) \cup \ldots \cup (s_{h-1}^{\tau_2}, a_{h-1}^{\tau_2}, s_h^{\tau_2})$
    $T = T \cup (\tau_1, \tau_2, b)$
    FEEDBACK$(H, T, f, g)$
end while
```

Given a policy π, we can define the standard RL value function as the expected total utility of being in state s at step i, and following policy π:

$$V_{\pi,i}(s) = \mathbb{E}\left[\sum_{j=i}^{h} \overline{r}(s_j, \pi(s_j)) | s_i = s\right], \tag{1}$$

and now we can define the optimal policy π^* as the one with maximal value for all input states. Note that $\mathbb{E}_{s_0 \sim p_0}[V_{\pi,0}(s_0)] \equiv \mathbb{E}_{\tau \sim \pi, \mathcal{M}}[\overline{r}(\tau)]$. Given fully specified dynamics and reward models, p and $\overline{r}$, it is straightforward to apply standard dynamic programming approaches such as value iteration to arrive at the optimal policy under p and $\overline{r}$ (Sutton & Barto, 2018). The goal of learning, then, is infer p and $\overline{r}$ to the extent necessary for good decision-making.

Learning Problem. In each iteration (or episode) i, the agent selects two policies, π_{i1} and π_{i2}. The two policies are rolled out to obtain trajectories τ_{i1} and τ_{i2}, and a binary preference $b_i \in \{0, 1\}$ between them is sampled according to the underlying utilities of τ_{i1} and τ_{i2}. We quantify the performance of the learning agent using expected cumulative regret relative to the optimal policy:

$$\mathbb{E}[\text{REG}_T] = \mathbb{E}\Bigg\{\sum_{i=1}^{\lceil T/(2h) \rceil} \sum_{s \in \mathcal{S}} p(s)\Big[2V_{\pi^*,0}(s) \tag{2}$$

$$- V_{\pi_{i1},0}(s) - V_{\pi_{i2},0}(s)\Big]\Bigg\}. \tag{3}$$

To minimize regret, the agent must balance exploration (collecting new data) with exploitation (behaving optimally w.r.t. existing models). In contrast to the standard formulation in RL (Osband et al., 2013), at each iteration/episode we compare the utilities of both selected policies.

3. Algorithm

As outlined in Algorithm 1, Dueling Posterior Sampling (DPS) iterates over three main steps: (a) sample two policies π_1, π_2 from the Bayesian posteriors of the dynamics and utility models (Advance – Algorithm 2); (b)

Algorithm 2 ADVANCE: Sample policy from dynamics and utility models

```
Input: $H, T, f, g$
Sample $M \sim f(\cdot|H)$ {Sample MDP transition dynamics from posterior}
Sample $R \sim g(\cdot|T)$ {Sample utilities from posterior}
Compute $\pi = \text{argmax}_\pi V(M, R)$ {Value iteration yields sampled MDP's optimal policy}
Return $\pi$
```

Algorithm 3 FEEDBACK: Update dynamics and utility models based on new user feedback

```
Input: $H, T, f, g$
Apply Bayesian update to $f$, given $H$ {Update dynamics model given history}
Apply Bayesian update to $g$, given $T$ {Update utility model given preferences}
Return $f$, $g$
```

roll out π_1 and π_2 to obtain trajectories τ_1 and τ_2, and receive preference feedback between them; (c) store the new state transitions and feedback and update the posterior (FEEDBACK – Algorithm 3). Compared to conventional posterior sampling with absolute feedback (Osband et al., 2013), the two key differences are that: two policies are sampled rather than one each iteration, and a credit assignment problem is solved when learning from feedback.

ADVANCE (Algorithm 2) samples from the Bayesian posteriors of the dynamics and utility models. The sampled dynamics and utilities form an MDP, and value iteration is used to derive the optimal policy π under the sample. Intuitively, peaked (i.e., certain) posteriors lead to less variability when sampling π, which implies less exploration, while diffuse (i.e., uncertain) posteriors lead to greater variability when sampling π, which implies more exploration.

FEEDBACK (Algorithm 3) updates the Bayesian posteriors of the dynamics and utility models based on new data. Updating the dynamics posterior is relatively straightforward, as we assume the dynamics are fully-observed; for instance, the dynamics prior can be modeled via Dirichlet distributions with multinomial conjugate observation likelihoods (Osband et al., 2013). In contrast, performing Bayesian inference over state/action utilities from trajectory-level feedback is much more challenging. Considering a range of approaches (see Appendix A1), we found Bayesian logistic regression (Section 3.1) to both be well-performing and admit tractable analysis within our theoretical framework.

3.1. Bayesian Logistic Regression for Utility Inference and Credit Assignment

Credit assignment (Wirth, 2017) is the problem of inferring which state/action pairs are responsible for observed trajectory-level preferences. We detail a Bayesian logistic regression approach to address this task in our setting. Logistic regression is a binary classification method that learns a weight vector $\boldsymbol{w}$ for the model $p(y = 1|\boldsymbol{x}, \boldsymbol{w}) = \frac{1}{1+\exp(-\boldsymbol{w}^T\boldsymbol{x})}$. Bayesian logistic regression (Albert & Chib, 1993; Murphy, 2012) maintains a posterior over possible weight vectors. Because there is no convenient prior yielding a closed-form conjugate posterior, we use the Laplace approximation to the posterior as specified below.

Preliminaries. Let N be the number of trajectories pairs observed so far, and $D = SA$ be the total number of state/action pairs. Let $\boldsymbol{x}_{ij} \in \mathbb{R}^D, j \in \{1, 2\}$ be the visitation vector corresponding to trajectory τ_{ij}, with the k^{th} element $\boldsymbol{x}_{ij}^{(k)}$ being the number of times that state/action pair k was visited in τ_{ij}. Define $\boldsymbol{x}_i := \boldsymbol{x}_{i1} - \boldsymbol{x}_{i2}$. The observation matrix X and label vector $\boldsymbol{y}$ are defined as:

$$X = \begin{bmatrix} (\boldsymbol{x}_{11} - \boldsymbol{x}_{12})^T \\ \vdots \\ (\boldsymbol{x}_{N1} - \boldsymbol{x}_{N2})^T \end{bmatrix}, \boldsymbol{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}, \tag{4}$$

where $y_i = 2\mathbb{I}_{[\tau_{i1} > \tau_{i2}]} - 1$, so that $y_i \in \{-1, 1\}$.

The observation matrix $X \in \mathbb{R}^{N \times D}$ has rank at most $D - 1$, since each row $\boldsymbol{x}_i = \boldsymbol{x}_{i1} - \boldsymbol{x}_{i2}$ must sum to zero. To obtain a full-row-rank observation matrix for Bayesian logistic regression, we transform $X \in \mathbb{R}^{N \times D}$ to $W \in \mathbb{R}^{N \times (D-1)}$ via the matrix $V = \begin{bmatrix} \boldsymbol{v}_1 & \dots & \boldsymbol{v}_{D-1} \end{bmatrix} \in \mathbb{R}^{D \times (D-1)}$, where $\boldsymbol{v}_i \in \mathbb{R}^D$ form an orthonormal basis spanning the $(D-1)$-dimensional, full *possible* row space of X. To obtain the vector $\boldsymbol{w}_i \in \mathbb{R}^{D-1}$ that expresses $\boldsymbol{x}_i$ in this basis, apply:

$$\boldsymbol{w}_i = [\boldsymbol{x}_i^T \boldsymbol{v}_1 \dots \boldsymbol{x}_i^T \boldsymbol{v}_{D-1}]^T = V^T \boldsymbol{x}_i, \tag{5}$$

while $\boldsymbol{w}_i \in \mathbb{R}^{D-1}$ is converted to the original space via:

$$\boldsymbol{x_i} = \sum_{j=1}^{D-1} w_{ij} \boldsymbol{v}_j = V \boldsymbol{w}_i, \tag{6}$$

where w_{ij} is the j^{th} element of $\boldsymbol{w}_i$.

Utility Model & Posterior Inference. We fit a Bayesian logistic regression model to the transformed data $(W, \boldsymbol{y})$. This model predicts the probability that τ is preferred to τ' as a logistic regression function of their visitation vector differences $\boldsymbol{x}_\tau - \boldsymbol{x}_{\tau'}$. The model parameters correspond exactly to the state/action utilities $\overline{r}$. The model internally computes an element-wise product between $\boldsymbol{x}_\tau - \boldsymbol{x}_{\tau'}$ and estimated reward vector $\boldsymbol{r}$, within the $(D-1)$-dimensional space given by (5). Because (5) preserves inner products (see Appendix A2), this is exactly the trajectory utility, and taking the expectation over trajectories generated by a policy is exactly the value function (1).

We are chiefly interested in sampling from the posterior of parameter/utility vector $\overline{\boldsymbol{r}}$, which can be combined with the sampled dynamics to perform value iteration and obtain a policy. As shown below, using the Laplace approximation, the posterior is Gaussian distributed, from which samples can easily be drawn. The internal utility representation lies in $\tilde{\boldsymbol{r}}' \in \mathbb{R}^{D-1}$, and we convert to $\tilde{\boldsymbol{r}} \in \mathbb{R}^D$ via (6).

We now describe the Bayesian logistic regression step. A Gaussian prior is defined over utilities $\boldsymbol{r}' \in \mathbb{R}^{D-1}$: $p(\boldsymbol{r}') \sim \mathcal{N}(\boldsymbol{r}'|\boldsymbol{r}_0, V_0)$. The logistic regression likelihood is:

$$p(W, \boldsymbol{y}|\boldsymbol{r}') = \prod_{i=1}^{N} \frac{1}{1 + \exp(y_i \boldsymbol{w}_i^T \boldsymbol{r}')}. \tag{7}$$

We model the posterior via the Laplace approximation:

$$p(\boldsymbol{r}'|W, \boldsymbol{y}) \approx \mathcal{N}(\boldsymbol{r}'|\hat{\boldsymbol{r}}', H^{-1}), \text{ where:} \tag{8}$$

$$\hat{\boldsymbol{r}}' = \operatorname*{argmin}_{\boldsymbol{r}'} f(\boldsymbol{r}'), \tag{9}$$

$$f(\boldsymbol{r}') := -\log p(\boldsymbol{r}') - \log p(W, \boldsymbol{y}|\boldsymbol{r}'), \tag{10}$$

$$H = \nabla^2_{\boldsymbol{r}'} f(\boldsymbol{r}')\Big|_{\hat{\boldsymbol{r}}'}. \tag{11}$$

To show a regret convergence using this approximate posterior, we leverage asymptotic normality of the maximum likelihood estimator of logistic regression in our proofs.

4. Theoretical Results

We now sketch our asymptotic no-regret analysis for DUELING POSTERIOR SAMPLING (DPS) with Bayesian logistic regression; the full proof is in Appendix A2. Additionally, Appendix A2.1 discusses possible avenues for extending this proof methodology toward analyzing other credit assignment models. The proof has two main parts: first proving that DPS with logistic credit assignment is asymptotically consistent (Theorem 1), and then proving that DPS has a sublinear regret rate (Theorem 2). Both parts leverage results on the asymptotic behavior of logistic regression (Gourieroux & Monfort, 1981). As before, we consider Bayesian logistic regression with data $W \in \mathbb{R}^{N \times (D-1)}$ and labels $\boldsymbol{y} \in \mathbb{R}^N$, with $[W]_{ij} = w_{ij}$. To show that DPS is asymptotically consistent in learning the reward function, we first provide some definitions and necessary conditions.

Definition 1 (Derivative of sigmoid). *$f : \mathbb{R} \longrightarrow \mathbb{R}$, where $f = \frac{e^{-x}}{(1+e^{-x})^2}$. Note that $f(x) = f(-x)$.*

Definition 2. *Let $\overline{\boldsymbol{r}} \in \mathbb{R}^{D-1}$ be the vector of true state/action utilities; we assume $\overline{\boldsymbol{r}}$ exists. Define $\tilde{\boldsymbol{r}}_k \in \mathbb{R}^{D-1}$ as the state/action rewards sampled from the posterior in episode k, $\hat{\boldsymbol{r}}_k \in \mathbb{R}^{D-1}$ as the maximum a posteriori (MAP) estimate of the Bayesian logistic regression model at episode k, and finally, $\hat{\boldsymbol{r}}_{ML,k} \in \mathbb{R}^{D-1}$ as the maximum likelihood estimate of the logistic regression model at k.*

Condition 1. *$\exists\, M_0$ such that $|w_{ij}| \leq M_0$ for all $i \in \{1, \dots, N\}, j \in \{1, \dots, D-1\}$.*

Condition 2. *Let $\lambda_1^{(k)}$ and $\lambda_{D-1}^{(k)}$ be the largest and smallest eigenvalues, respectively, of $\sum_{i=1}^k f(\boldsymbol{w}_i^T \overline{\boldsymbol{r}}) \boldsymbol{w}_i \boldsymbol{w}_i^T$. Then, $\exists\, M_1$ such that $\frac{\lambda_1^{(k)}}{\lambda_{D-1}^{(k)}} < M_1$, for all k.*

Proposition 1 (Asymptotic consistency of logistic regression (Gourieroux & Monfort, 1981))**.** *If Conditions 1 and 2 are satisfied, then the maximum likelihood estimator $\hat{\boldsymbol{r}}_{ML,k}$ of $\overline{\boldsymbol{r}}$ exists almost surely as $k \longrightarrow \infty$, and $\hat{\boldsymbol{r}}_{ML,k}$ converges almost surely to the true values $\overline{\boldsymbol{r}}$ if and only if $\lim_{k \longrightarrow \infty} \lambda_{D-1}^{(k)} = \infty$.*

We first show that Proposition 1's final condition is satisfied with known transition dynamics, and afterwards consider the convergence of the dynamics model posterior.

Lemma 1. *Under known transition dynamics, all eigenvalues of the matrix $\sum_{i=1}^k f(\boldsymbol{w}_i^T \overline{\boldsymbol{r}}) \boldsymbol{w}_i \boldsymbol{w}_i^T$ approach infinity as $k \longrightarrow \infty$.*

Lemma 2 (Convergence of dynamics model)**.** *Given Lemma 1, DPS's dynamics model converges to the true dynamics, and as it converges, all eigenvalues of $\sum_{i=1}^k f(\boldsymbol{w}_i^T \overline{\boldsymbol{r}}) \boldsymbol{w}_i \boldsymbol{w}_i^T$ approach infinity.*

Combining these results, we obtain:

Theorem 1 (Asymptotic consistency of DPS)**.** *If there exists a reward function such that a logistic regression model explains the preference feedback, then DPS with a Bayesian logistic regression credit assignment model will learn an asymptotically consistent reward model.*

We turn next to characterizing the regret rate of DPS. We apply two prior results, one from Gourieroux and Monfort (1981) regarding the asymptotic distribution of the logistic regression maximum likelihood estimate (Prop. 2), and the other from Osband et al. (2013) regarding a regret bound for posterior sampling RL (Prop. 3).

Proposition 2 (Asymptotic normality of logistic regression maximum likelihood estimator (Gourieroux & Monfort, 1981))**.** *If Conditions 1 and 2 are satisfied, and if $\hat{\boldsymbol{r}}_{ML,k}$ converges almost surely to the true value $\overline{\boldsymbol{r}}$, then:*

$$\left[\sum_{i=1}^{k} f(\boldsymbol{w}_i^T \hat{\boldsymbol{r}}_{ML,k}) \boldsymbol{w}_i \boldsymbol{w}_i^T\right]^{\frac{1}{2}} (\hat{\boldsymbol{r}}_{ML,k} - \overline{\boldsymbol{r}}) \tag{12}$$

$$\xrightarrow{D} \mathcal{N}(\boldsymbol{0}, \mathbb{I}) \text{ as } k \longrightarrow \infty, \tag{13}$$

where $\xrightarrow{D}$ implies convergence in distribution and $Q^{\frac{1}{2}}$ is the positive definite matrix associated with positive definite matrix Q such that $[Q^{\frac{1}{2}}]^2 = Q$.

Proposition 3 (Expected regret of posterior sampling RL (Osband et al., 2013))**.** *Posterior sampling RL has expected T-step regret $O(hS\sqrt{ATlog(SAT)})$, with horizon h and numbers of states and actions S and A.*

Leveraging these results, we show that under preference feedback, the regret can be decomposed into two terms: one that reflects the converging dynamics model, and one that reflects the converging reward model (inferred from trajectory-level preference feedback).

Lemma 3 (Regret decomposition)**.** *The expected regret of* DPS *can be decomposed into two terms. One term can be bounded by the regret bound of Osband et al. (2013), stated in Proposition (3). The other is bounded by:* $h\sum_{k=1}^{\lceil T/h \rceil} \mathbb{E}[||\overline{\boldsymbol{r}} - \tilde{\boldsymbol{r}}_k||_\infty] \leq h\sum_{k=1}^{\lceil T/h \rceil} \mathbb{E}[||\hat{\boldsymbol{r}}_k - \overline{\boldsymbol{r}}||_\infty] + h\sum_{k=1}^{\lceil T/h \rceil} \mathbb{E}[||\hat{\boldsymbol{r}}_{\boldsymbol{k}} - \tilde{\boldsymbol{r}}_k||_\infty]$.

The most burdensome part of our proof is analyzing convergence of $\tilde{\boldsymbol{r}}_k$ to $\hat{\boldsymbol{r}}_k$ and $\hat{\boldsymbol{r}}_k$ to $\overline{\boldsymbol{r}}$, which requires analyzing convergence of credit assignment. Afterwards, we reach the final result:

Theorem 2 (Asymptotic regret rate of DPS)**.** *If there exists a reward function such that a logistic regression model explains the preference feedback, then* DPS *has an asymptotic no-regret rate of* $O\left(hS\sqrt{ATlog(SAT)} + h\sqrt{\frac{SA}{c_0}Tlog(T)}\right)$*, where* c_0 *is a minimum linear rate at which all eigenvalues of* $\sum_{i=1}^{k} f(\boldsymbol{w}_i^T \overline{\boldsymbol{r}})\boldsymbol{w}_i\boldsymbol{w}_i^T$ *must increase with k.*

5. Experiments

We validate the empirical performance of DUELING POSTERIOR SAMPLING (DPS) on two simulated domains with varying preference functions, and also evaluate DPS using alternative credit assignment models. We find that DPS generally performs well, and outperforms standard PBRL baselines (Wirth & Fürnkranz, 2013a).

Experimental Setup. We evaluate on two simulated environments: RiverSwim and random MDPs. The RiverSwim environment (Osband et al., 2013) has six states and two actions (actions 0 and 1); the optimal policy is to always choose action 1, which maximizes the probability of reaching a particular goal state/action pair. Meanwhile, a suboptimal policy—yielding a much smaller reward compared to the goal—is quickly and easily discovered and incentivizes the agent to always select action 0. The algorithm must demonstrate sufficient exploration to have hope of discovering the optimal policy quickly.

In the second simulated environment, we generate random MDPs according to the procedure of Osband et al. (2013). Each random MDP is generated with 50 states and 5 actions, and the transition dynamics and rewards are generated from Dirichlet and Normal-Gamma distributions, respectively. All parameters of these two distributions were set to 1 to obtain a diffuse distribution over possible MDPs. The sampled reward vectors were shifted and normalized so that all rewards fell between 0 and 1.

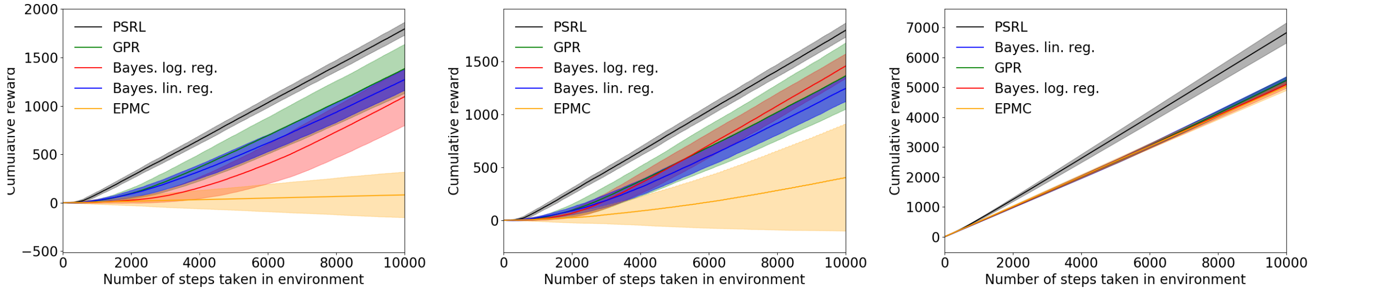
In both of these environments, preferences between pairs of trajectories were generated by (noisily) comparing the total rewards that they accumulated; this reward information was hidden from the learning algorithm, which observed only the trajectory preferences and state transitions. Preference noise is generated according to a logistic model: for trajectories τ_i and τ_j, $P(\tau_i > \tau_j) = \{1 + \exp[-c(\overline{r}(\tau_i) - \overline{r}(\tau_j))]\}^{-1}$, where $\overline{r}(\tau_i)$ and $\overline{r}(\tau_j)$ are the total rewards accrued by the two trajectories, respectively, while the hyperparameter c controls the degree of noisiness.

Methods Compared. We evaluate DPS under three noise levels ($c \in \{0.1, 1, 1000\}$) and three credit assignment models: 1) Bayesian logistic regression, 2) Bayesian linear regression, and 3) Gaussian process regression, where the latter two are described in Appendix A1. In addition, we evaluate Every-Visit Preference Monte Carlo (EPMC) with probabilistic credit assignment (Wirth & Fürnkranz, 2013b; Wirth, 2017) as a baseline. Lastly, we compare against the posterior sampling RL algorithm (Osband et al., 2013), which learns using the true numerical rewards at each step, and thus yields an upper-bound on the performance that a preference-based algorithm could achieve.

Results. Figure 1 shows the performance comparison for $c = 1$ in both environments, as well as $c = 1,000$ in RiverSwim (additional results are in Appendix A3, including hyperparameter details). DPS performs well in all simulations, and sometimes significantly outperforms the EPMC baseline. This may be because EPMC uses a uniform exploration strategy, while DPS prioritizes exploration by sampling high rewards for more uncertain state/action pairs. Notice that $c = 1,000$ results in nearly-noiseless preferences; this can decrease performance in RiverSwim in some cases, since preference noise can help the agent to escape the local minimum. We also see that DPS is competitive with PSRL, which has access to the full cardinal rewards at each state/action. Finally, we see that the performance of DPS is robust to the choice of credit assignment model, and in fact using Gaussian process regression (for which we do not have an end-to-end regret analysis) often leads to the best empirical performance. These results suggest that DPS is a practically promising approach that can robustly incorporate many modeling approaches as subroutines.

6. Conclusion

We investigate the preference-based reinforcement learning problem, which receives comparative preferences instead of absolute real-valued rewards as feedback. We develop the DUELING POSTERIOR SAMPLING (DPS) algorithm, which optimizes policies in an highly efficient and flexible way. To our knowledge, DPS is the first preference-based RL algorithm with a regret guarantee. DPS also performs well in our simulations, and seems practically promising.



(a) RiverSwim, $c = 1,000$

(b) RiverSwim, $c = 1$

(c) Random MDP, $c = 1$

Figure 1. Empirical performance of DPS. a) and b) show RiverSwim with noise hyperparameters $c = 1,000, 1$. c) displays random MDPs with $c = 1$. Posterior sampling RL (PSRL) (Osband et al., 2013) is an upper-bound that receives numerical rewards; Gaussian process regression (GPR), Bayesian linear regression, and Bayesian logistic regression are all instances of DPS. EPMC is a baseline from Wirth and Fürnkranz (2013b) as discussed. Plots display mean +/- one std over 100 runs of each algorithm tested. Additional results (more values of c) are in Appendix A3. Overall, we see that DPS performs well and is robust to the choice of credit assignment model.

There are many directions for future work. The Bayesian logistic regression model could be improved with more accurate posterior estimates. Assumptions governing the user's preferences, such as requiring an underlying utility model, could be relaxed. One can also incorporate kernelized methods to further improve sample efficiency. It is also important to extend to other credit assignment models, such as the Gaussian process regression and Bayesian linear regression methods, for which the same concept of the regret decomposition still applies. We expect that DPS would perform well with any asymptotically consistent reward model that sufficiently captures users' preference behavior.

Acknowledgements

This work was funded by NIH grant EB007615 and an Amazon graduate fellowship.

References

Agrawal, S. and Jia, R. Optimistic posterior sampling for reinforcement learning: Worst-case regret bounds. In *Advances in Neural Information Processing Systems*, pp. 1184–1194, 2017.

Akrour, R., Schoenauer, M., and Sebag, M. APRIL: Active preference learning-based reinforcement learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 116–131. Springer, 2012.

Albert, J. H. and Chib, S. Bayesian analysis of binary and polychotomous response data. *Journal of the American statistical Association*, 88(422):669–679, 1993.

Argall, B. D., Chernova, S., Veloso, M., and Browning, B. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.

Basu, C., Yang, Q., Hungerman, D., Sinahal, M., and Dragan, A. D. Do you want your autonomous car to drive like you? In *2017 12th ACM/IEEE International Conference on Human-Robot Interaction (HRI*, pp. 417–425. IEEE, 2017.

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, pp. 4299–4307, 2017.

Dudík, M., Hofmann, K., Schapire, R. E., Slivkins, A., and Zoghi, M. Contextual dueling bandits. In *Conference on Learning Theory (COLT)*, 2015.

Gopalan, A. and Mannor, S. Thompson sampling for learning parameterized Markov decision processes. In *Conference on Learning Theory*, pp. 861–898, 2015.

Gourieroux, C. and Monfort, A. Asymptotic properties of the maximum likelihood estimator in dichotomous logit models. *Journal of Econometrics*, 17(1):83–97, 1981.

Kupcsik, A., Hsu, D., and Lee, W. S. Learning dynamic robot-to-human object handover from human feedback. In *Robotics research*, pp. 161–176. Springer, 2018.

Murphy, K. P. *Machine learning: A probabilistic perspective*. The MIT Press, 2012.

Osband, I. and Van Roy, B. Why is posterior sampling better than optimism for reinforcement learning? In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2701–2710. JMLR. org, 2017.

Osband, I., Russo, D., and Van Roy, B. (More) efficient reinforcement learning via posterior sampling. In *Advances in Neural Information Processing Systems*, pp. 3003–3011, 2013.

Sadigh, D., Dragan, A. D., Sastry, S., and Seshia, S. A. Active preference-based learning of reward functions. In *Robotics: Science and Systems (RSS)*, 2017.

Sui, Y., Zhuang, V., Burdick, J. W., and Yue, Y. Multi-dueling bandits with dependent arms. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2017.

Sui, Y., Zhuang, V., Burdick, J., and Yue, Y. Stagewise safe Bayesian optimization with Gaussian processes. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4781–4789. PMLR, 10–15 Jul 2018a.

Sui, Y., Zoghi, M., Hofmann, K., and Yue, Y. Advancements in dueling bandits. In *IJCAI*, pp. 5502–5510, 2018b.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.

Szörényi, B., Busa-Fekete, R., Paul, A., and Hüllermeier, E. Online rank elicitation for Plackett-Luce: A dueling bandits approach. In *Advances in Neural Information Processing Systems*, pp. 604–612, 2015.

Thompson, W. R. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

Wirth, C. *Efficient Preference-based Reinforcement Learning*. PhD thesis, Technische Universität, 2017.

Wirth, C. and Fürnkranz, J. EPMC: Every visit preference Monte Carlo for reinforcement learning. In *Asian Conference on Machine Learning*, pp. 483–497, 2013a.

Wirth, C. and Fürnkranz, J. A policy iteration algorithm for learning from preference-based feedback. In *International Symposium on Intelligent Data Analysis*, pp. 427–437. Springer, 2013b.

Wirth, C., Fürnkranz, J., and Neumann, G. Model-free preference-based reinforcement learning. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

Wirth, C., Akrour, R., Neumann, G., and Fürnkranz, J. A survey of preference-based reinforcement learning methods. *The Journal of Machine Learning Research*, 18(1): 4945–4990, 2017.

Wu, H. and Liu, X. Double Thompson sampling for dueling bandits. In *Advances in Neural Information Processing Systems*, pp. 649–657, 2016.

Yue, Y., Broder, J., Kleinberg, R., and Joachims, T. The k-armed dueling bandits problem. *Journal of Computer and System Sciences*, 78(5):1538–1556, 2012.

Zoghi, M., Whiteson, S., Munos, R., and De Rijke, M. Relative upper confidence bound for the k-armed dueling bandit problem. In *International Conference on Machine Learning (ICML)*, 2014.