Dueling Posterior Sampling for Preference-Based Reinforcement Learning

Ellen R. Novoseller[1], Yibing Wei[1], Yanan Sui[2], Yisong Yue[1], Joel W. Burdick[1]
[1]Department of Computing and Mathematical Sciences, California Institute of Technology, Pasadena, CA 91125
[2] School of Aerospace Engineering, Tsinghua University, Beijing, China 100084
{enovoseller, ywwei, yyue}@caltech.edu, ysui@tsinghua.edu.cn, jwb@robotics.caltech.edu

Abstract

In preference-based reinforcement learning (RL), an agent interacts with the environment while receiving preferences instead of absolute feedback. While there is increasing research activity in preference-based RL, the design of formal frameworks that admit tractable theoretical analysis remains an open challenge. Building upon ideas from preference-based bandit learning and posterior sampling in RL, 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 preference feedback. As preference feedback is provided on trajectories rather than individual state-action pairs, we develop a Bayesian approach for the credit assignment problem, translating preferences to a posterior distribution over state-action reward models. We prove an asymptotic Bayesian no-regret rate for DPS with a Bayesian linear regression credit assignment model. This is the first regret guarantee for preference-based RL to our knowledge. We also discuss possible avenues for extending the proof methodology to other credit assignment models. Finally, we evaluate the approach empirically, showing competitive performance against existing baselines.

1 INTRODUCTION

Reinforcement learning (RL) agents interact with humans in many domains, from clinical trials (Sui et al., 2018a) to autonomous driving (Sadigh et al., 2017) to human-robot interaction (Kupcsik et al., 2018), and take human preferences as feedback. While many RL algorithms assume the existence of a numerical reward signal, in settings involving humans, it is often unclear how to define a reward signal that accurately reflects optimal system-human interaction. For instance, in autonomous driving (Basu et al., 2017) and robotics (Argall et al., 2009; Akrour et al., 2012), users can have difficulty with both specifying numerical reward functions and providing demonstrations of desired behavior. Moreover, a misspecified reward function can result in "reward hacking" (Amodei et al., 2016), in which undesirable actions achieve high rewards. In such situations, the user's *preferences* could more reliably measure her intentions.

This work studies the problem of preference-based reinforcement learning (PBRL), in which the RL agent executes a pair of trajectories of interaction with the environment, and the user provides (noisy) pairwise preference feedback, revealing which of the two trajectories is preferred. Though 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. While the preference-based bandit setting—in which the agent observes preferences between selected actions—has seen significant theoretical progress (e.g., Yue et al. (2012); Zoghi et al. (2014); Ailon et al. (2014); Szörényi et al. (2015); Dudík et al. (2015); Zoghi et al. (2015); Ramamohan et al. (2016); Wu and Liu (2016); Sui et al. (2017, 2018b)), the PBRL setting is more challenging, as the environment's dynamics can stochastically translate the agent's policies for interaction (analogous to actions in the bandit setting) to the observed trajectories.

In this paper, we present the DUELING POSTERIOR SAMPLING (DPS) algorithm, which uses preference-based posterior sampling to tackle PBRL in the Bayesian regime. Posterior sampling (Thompson, 1933), also called Thompson sampling, is a Bayesian model-based approach to balancing exploration and exploitation, which enables the algorithm to efficiently learn models

Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI), PMLR volume 124, 2020.

of both the environment's state transition dynamics and reward function. Previous work on posterior sampling in RL (Osband et al., 2013; Gopalan and Mannor, 2015; Agrawal and Jia, 2017; Osband and Van Roy, 2017) is focused on learning from absolute rewards, while we 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), but has a quite different theoretical analysis, as we need to incorporate trajectory-level preference learning and state transition dynamics.

Learning from trajectory-level preferences is in general a very challenging problem, as information about the rewards is sparse (often just one bit), is only relative to the pair of trajectories being compared, and does not explicitly include information about actions within trajectories. DPS learns from preference feedback by internally maintaining 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), i.e., determining which of the encountered states and actions are responsible for the trajectory-level preference feedback.

We developed DPS concurrently with an analysis framework for characterizing regret convergence in the episodic setting, based upon information-theoretic techniques for bounding the Bayesian regret of posterior sampling (Russo and Van Roy, 2016). We mathematically integrate Bayesian credit assignment and preference elicitation within the conventional posterior sampling framework, evaluate several credit assignment models, and prove a Bayesian asymptotic no-regret rate for DPS with a Bayesian linear regression credit assignment model. To our knowledge, this is the first PBRL approach with theoretical guarantees. We also demonstrate that DPS delivers competitive performance empirically.

2 RELATED WORK

Posterior sampling. Balancing exploration and exploitation is a key problem in RL. In the episodic learning setting, the agent typically aims to balance exploration and exploitation to minimize its regret, i.e., the gap between the expected total rewards of the agent and the optimal policy. Posterior sampling, first proposed in Thompson (1933), is a Bayesian model-based approach toward achieving this goal, which iterates between (1) updating the posterior of a Bayesian environment model and (2) sampling from this posterior to select the next policy. In both the bandit and RL settings, posterior sampling has been demonstrated to perform competitively in experiments and enjoy favorable theoretical regret guarantees (Osband and Van Roy, 2017; Osband et al., 2013; Agrawal and Jia, 2017; Chapelle and Li, 2011).

Our approach builds upon two existing posterior sampling algorithms: Self-Sparring (Sui et al., 2017) for preference-based bandit learning (also known as dueling bandits (Yue et al., 2012)) and posterior sampling RL (Osband et al., 2013). Self-Sparring maintains a posterior over each action's reward, and in each iteration, draws multiple samples from this posterior to "duel" or "spar" via preference elicitation. For each set of sampled rewards, the algorithm executes the action with the highest reward sample, obtaining new preferences to update the model posterior. Sui et al. (2017) prove an asymptotic no-regret guarantee for Self-Sparring with independent Beta-Bernoulli reward models for each action.

Within RL, posterior sampling has been applied to the finite-horizon setting with absolute rewards to learn Bayesian posteriors over both the dynamics and rewards. Each posterior sample yields models of both dynamics and rewards, which are used to compute the optimal policy for the sampled system. This policy is executed to get a roll-out trajectory, used to update the dynamics and reward posteriors. In Osband et al. (2013), the authors show an expected regret of $O(hS\sqrt{AT\log(SAT)})$ after T time-steps, with finite time horizon h and discrete state and action spaces of sizes S and A, respectively.

Our theoretical analysis studies the Bayesian linear regression credit assignment model, which most closely resembles Bayesian reward modeling in the linear bandit setting (Abbasi-Yadkori et al., 2011; Agrawal and Goyal, 2013; Abeille and Lazaric, 2017). While both the PBRL and linear bandit settings apply Bayesian linear regression to recover model parameters, PBRL additionally requires learning the dynamics, determining policies via value iteration, and receiving feedback as preferences between trajectory pairs.

Several regret analyses in the linear bandit domain (Abbasi-Yadkori et al., 2011; Agrawal and Goyal, 2013; Abeille and Lazaric, 2017) rely upon martingale concentration properties introduced in Abbasi-Yadkori et al. (2011), and depend upon a bound that is not applicable in the preference-based setting (see Appendix C). Intuitively, these analyses assume that the agent learns about rewards with respect to every observation's feature vector. In contrast, the preference-based setting assumes that only the *difference* in the total rewards of two trajectories affects human preferences. Thus, while the algorithm incurs regret with respect to every sampled trajectory, only

differences between compared trajectory feature vectors yield information about rewards.

Our regret analysis takes inspiration from the information-theoretic perspective on Thompson sampling introduced in Russo and Van Roy (2016), a framework for quantifying Bayesian regret in terms of the information gained at each step about the optimal action. This analysis focuses upon upper-bounding the *information ratio*, which quantifies the trade-off between exploration (via the information gain) and exploitation (via the instantaneous regret) at each step. Several studies (Zanette and Sarkar, 2017; Nikolov et al., 2018) consider extensions of this work to the RL setting, but to our knowledge, it has not previously been applied toward preference-based learning.

Preference-based learning. Previous work on PBRL has shown successful performance in a number of applications, including Atari games and the Mujoco environment (Christiano et al., 2017), learning human preferences for autonomous driving (Sadigh et al., 2017), and selecting a robot's controller parameters (Kupcsik et al., 2018; Akrour et al., 2014). Yet, to our knowledge, the PBRL literature still lacks theoretical guarantees.

Much of the existing work in PBRL handles a distinct setting from ours. While we seek online regret minimization, several existing algorithms minimize the number of preference queries (Christiano et al., 2017; Wirth et al., 2016). Such algorithms, for instance those which apply deep learning, typically assume that many simulations can be cheaply run between preference queries. In contrast, our setting assumes that experimentation is as expensive as preference elicitation; this could include such domains as adaptive experiment design and human-robot interaction without well-understood human dynamics.

Existing approaches for trajectory-level preference-based RL may be broadly divided into three categories (Wirth, 2017): a) directly optimizing policy parameters (Wilson et al., 2012; Busa-Fekete et al., 2013; Kupcsik et al., 2018); b) modeling action preferences in each state (Fürnkranz et al., 2012); and c) learning a utility function to characterize the rewards, returns, or values of state-action pairs (Wirth and Fürnkranz, 2013a,b; Akrour et al., 2012; Wirth et al., 2016; Christiano et al., 2017). In c), the utility is often modeled as linear in the trajectory features. If those features are defined in terms of visitations to each state-action pair, then utility directly corresponds to the total (undiscounted) reward.

We adopt the third of these paradigms: PBRL with underlying utility functions. By inferring state-action rewards from preference feedback, one can derive relatively-interpretable reward models and employ such methods as value iteration. In addition, utility-based approaches may be more sample efficient compared to policy search and preference relation methods (Wirth, 2017), as they extract more information from each observation. Notably, Wilson et al. (2012) learn a Bayesian model over policy parameters, and sample from its posterior to inform actions. From existing PBRL methods, their algorithm perhaps most resembles ours; however, compared to utility-based approaches, policy search methods typically require either more samples or expert knowledge to craft the policy parameters (Wirth et al., 2017; Kupcsik et al., 2018).

Beyond RL, preference-based learning has been the subject of much research. The bandit setting (Yue et al., 2012; Zoghi et al., 2014; Ailon et al., 2014; Szörényi et al., 2015; Dudík et al., 2015; Zoghi et al., 2015; Ramamohan et al., 2016; Wu and Liu, 2016; Sui et al., 2017, 2018b) is closest, as it is essentially a single-state variant of RL. Other settings include: active learning (Sadigh et al., 2017; Houlsby et al., 2011; Eric et al., 2008), which is focused exclusively on learning an accurate model rather than maximizing utility of decision-making; learning with more structured preference feedback (Radlinski and Joachims, 2005; Shivaswamy and Joachims, 2012; Raman et al., 2013; Shivaswamy and Joachims, 2015), where the learner receives more than one bit of information per preference elicitation; and batch supervised settings such as learning to rank (Herbrich et al., 1999; Chu and Ghahramani, 2005; Joachims, 2005; Burges et al., 2005; Yue et al., 2007; Burges et al., 2007; Liu, 2009).

3 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, $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \phi, p, p_0, h)$, where the state space $\mathcal{S}$ and action space $\mathcal{A}$ are finite sets with cardinalities S and A, respectively. The agent episodically interacts with the environment in length-h roll-out trajectories of the form $\tau = \{s_1, a_1, s_2, a_2, \ldots, s_h, a_h, s_{h+1}\}$. In the i^{th} iteration, the agent executes two trajectory roll-outs τ_{i1} and τ_{i2} and observes a preference between them; we use the notation $\tau \succ \tau'$ to indicate a preference for trajectory τ over τ'. The initial state is sampled from p_0, while p defines the transition dynamics: $s_{t+1} \sim p(\cdot|s_t, a_t)$. Finally, the function ϕ captures the preference feedback generation mechanism: $\phi(\tau, \tau') := P(\tau \succ \tau') \in [0, 1]$.

A *policy*, $\pi : \mathcal{S} \times \{1, \ldots, h\} \longrightarrow \mathcal{A}$, is a (possibly-stochastic) mapping from states and time indices to actions. In each iteration i, the agent selects two policies, π_{i1} and π_{i2}, which are rolled out to obtain trajectories τ_{i1}

and τ_{i2} and preference label y_i. We represent each trajectory as a feature vector, where the features record the number of times each state-action pair is visited. In iteration i, rolled-out trajectories τ_{i1} and τ_{i2} correspond, respectively, to feature vectors $\boldsymbol{x}_{i1}, \boldsymbol{x}_{i2} \in \mathbb{R}^d$, where $d := SA$ is the total number of state-action pairs, and the k^{th} element of $\boldsymbol{x}_{ij}$, $j \in \{1, 2\}$, is the number of times that τ_{ij} visits state-action pair k. The preference for iteration i is denoted $y_i := \mathbb{I}_{[\tau_{i2} \succ \tau_{i1}]} - \frac{1}{2} \in \left\{-\frac{1}{2}, \frac{1}{2}\right\}$, where $\mathbb{I}_{[\cdot]}$ denotes the indicator function, so that $P\left(y_i = \frac{1}{2}\right) = 1 - P\left(y_i = -\frac{1}{2}\right) = \phi(\tau_{i2}, \tau_{i1}) - \frac{1}{2}$; there are no ties in any comparisons. Lastly, we define $\boldsymbol{x}_i := \boldsymbol{x}_{i2} - \boldsymbol{x}_{i1}$.

Our analysis builds upon two main assumptions. Firstly, we assume the existence of underlying utilities, quantifying the user's satisfaction with each trajectory:

Assumption 1. *Each trajectory τ has utility $\overline{r}(\tau)$, which decomposes additively: $\overline{r}(\tau) \equiv \sum_{t=1}^{h} \overline{r}(s_t, a_t)$ for the state-action pairs in τ. Defining $\overline{\boldsymbol{r}} \in \mathbb{R}^d$ as the vector of all state-action rewards, $\overline{r}(\tau)$ can also be expressed in terms of τ's state-action visit counts $\boldsymbol{x}$: $\overline{r}(\tau) = \overline{\boldsymbol{r}}^T \boldsymbol{x}$.*

Secondly, we assume that the utilities $\overline{r}(\tau)$ are stochastically translated to preferences via the noise model ϕ, such that the probability of observing $\tau_{i2} \succ \tau_{i1}$ is a function of the *difference* in their utilities. Intuitively, the greater the disparity in two trajectories' utilities, the more accurate the user's preference between them:

Assumption 2. *$P(\tau_{i2} \succ \tau_{i1}) = \phi(\tau_{i2}, \tau_{i1}) = g(\overline{r}(\tau_{i2}) - \overline{r}(\tau_{i1})) + \frac{1}{2} = g(\overline{\boldsymbol{r}}^T \boldsymbol{x}_{i2} - \overline{\boldsymbol{r}}^T \boldsymbol{x}_{i1}) + \frac{1}{2}$, where $g : \mathbb{R} \longrightarrow \left[-\frac{1}{2}, \frac{1}{2}\right]$ is a* link function *such that a) g is non-decreasing, and b) $g(x) = -g(-x)$ to ensure that $P(\tau \succ \tau') = 1 - P(\tau' \succ \tau)$. Note that if $\overline{r}(\tau) = \overline{r}(\tau')$, we have $P(\tau \succ \tau') = \frac{1}{2}$, and that $P(\tau_{i2} \succ \tau_{i1}) > \frac{1}{2} \Leftrightarrow g(\overline{\boldsymbol{r}}^T \boldsymbol{x}_i) > 0 \Leftrightarrow \overline{\boldsymbol{r}}^T \boldsymbol{x}_{i2} > \overline{\boldsymbol{r}}^T \boldsymbol{x}_{i1}$.*

For noiseless preferences, $g_{\text{ideal}}(x) := \mathbb{I}_{[x>0]} - \frac{1}{2}$. Alternatively, the logistic or Bradley-Terry link function is defined as $g_{\log}(x) := [1 + \exp(-x/c)]^{-1} - \frac{1}{2}$ with "temperature" $c \in (0, \infty)$. Our theoretical analysis assumes the linear link function (Ailon et al., 2014): $g_{\text{lin}}(x) := cx$, for $c > 0$ and $x \in [-\frac{1}{2c}, \frac{1}{2c}]$. Then, $\mathbb{E}[y_i] = P(\tau_{i2} \succ \tau_{i1}) - \frac{1}{2} = c\overline{\boldsymbol{r}}^T(\boldsymbol{x}_{i2} - \boldsymbol{x}_{i1})$. Without loss of generality, we set $c = 1$ by subsuming c into $\overline{\boldsymbol{r}}$. Denote the observation noise associated with g_{lin} on iteration i as η_i, such that $y_i = \overline{\boldsymbol{r}}^T(\boldsymbol{x}_{i2} - \boldsymbol{x}_{i1}) + \eta_i$.

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

$$V_{\pi,j}(s) = \mathbb{E}\left[\sum_{t=j}^{h} \overline{r}(s_t, \pi(s_t, t)) \,\middle|\, s_j = s\right]. \quad (1)$$

The optimal policy π^* is then defined as one that maximizes the expected value over all input states: $\pi^* = \sup_\pi \sum_{s \in \mathcal{S}} p_0(s) V_{\pi,1}(s)$. Note that $\mathbb{E}_{s_1 \sim p_0}[V_{\pi,1}(s_1)] \equiv \mathbb{E}_{\tau \sim (\pi, \mathcal{M})}[\overline{r}(\tau)]$. Given fully specified dynamics and rewards, 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 $\overline{r}$. The learning goal, then, is to infer p and $\overline{r}$ to the extent necessary for good decision-making.

Learning problem. We quantify the learning agent's performance via its cumulative T-step Bayesian regret relative to the optimal policy:

$$\mathbb{E}[\text{REG}(T)] = \mathbb{E}\Bigg\{ \sum_{i=1}^{\lceil T/(2h) \rceil} \sum_{s \in \mathcal{S}} p_0(s) \big[2V_{\pi^*,1}(s) - V_{\pi_{i1},1}(s) - V_{\pi_{i2},1}(s)\big] \Bigg\}. \quad (2)$$

To minimize regret, the agent must balance exploration (collecting new data) with exploitation (behaving optimally given current knowledge). Over-exploration of bad trajectories will incur large regret, and under-exploration can prevent convergence to optimality. In contrast to the standard regret formulation in RL, at each iteration we measure regret of both selected policies.

Assumptions. We make two further assumptions. The first imposes a regularity condition upon the noise η_i:

Assumption 3. *The label noise $\eta_i = y_i - \overline{\boldsymbol{r}}^T \boldsymbol{x}_i$ is conditionally R-sub-Gaussian, that is, there exists $R \geq 0$ such that $\forall\, \lambda \in \mathbb{R}$:*

$$\mathbb{E}\left[e^{\lambda \eta_i} \,\middle|\, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{i-1}, \eta_1, \ldots, \eta_{i-1}\right] \leq \exp\left(\frac{\lambda^2 R^2}{2}\right).$$

Note that bounded, zero-mean noise lying in an interval of length at most $2R$ is R-sub-Gaussian, and that sub-Gaussianity requires $\mathbb{E}[\eta_i \,|\, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_i, \eta_1, \ldots, \eta_{i-1}] = 0$ (Abbasi-Yadkori et al., 2011). Since $y_i \in \left\{-\frac{1}{2}, \frac{1}{2}\right\}$ and $\mathbb{E}[y_i \,|\, \boldsymbol{x}_i] = \overline{\boldsymbol{r}}^T \boldsymbol{x}_i \in \left[-\frac{1}{2}, \frac{1}{2}\right]$, we must have $\eta_i \in [-1, 1]$. Thus, η_i is R-sub-Gaussian with $R \leq 1$, provided that $\mathbb{E}[\eta_i \,|\, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_i, \eta_1, \ldots, \eta_{i-1}] = 0$. The latter holds by the assumption that $\mathbb{E}[y_i \,|\, \boldsymbol{x}_i] = \overline{\boldsymbol{r}}^T \boldsymbol{x}_i$.

Assumption 4. *For some known $S_r < \infty$, $||\overline{\boldsymbol{r}}||_2 \leq S_r$.*

Additional notation. For random variables X and X_n, $n \in \mathbb{N}$, $X_n \xrightarrow{D} X$ denotes that X_n converges to X in distribution. For $\boldsymbol{x} \in \mathbb{R}^d$ and positive definite matrix $B \in \mathbb{R}^{d \times d}$, we define the norm $||\boldsymbol{x}||_B := \sqrt{\boldsymbol{x}^T B \boldsymbol{x}}$.

4 ALGORITHM

As outlined in Algorithm 1, DUELING POSTERIOR SAMPLING (DPS) iterates among three steps: (a) sampling two policies π_{i1}, π_{i2} from the Bayesian posteriors

Algorithm 1 DUELING POSTERIOR SAMPLING (DPS)

$\mathcal{H}_0 = \emptyset$ {Initialize history}
Initialize prior for f_p {Initialize state transition model}
Initialize prior for f_r {Initialize utility model}
for $i = 1, 2, \ldots$ **do**
  $\pi_{i1} \leftarrow$ ADVANCE(f_p, f_r)
  $\pi_{i2} \leftarrow$ ADVANCE(f_p, f_r)
  Sample trajectories τ_{i1} and τ_{i2} from π_{i1} and π_{i2}
  Observe feedback $y_i = \mathbb{I}_{[\tau_{i2} \succ \tau_{i1}]} - \frac{1}{2}$
  $\mathcal{H}_i = \mathcal{H}_{i-1} \cup (\tau_{i1}, \tau_{i2}, y_i)$
  f_p, f_r = FEEDBACK($\mathcal{H}_i$, f_p, f_r)
end for

of the dynamics and utility models (ADVANCE – Algorithm 2); (b) rolling out π_{i1} and π_{i2} to obtain trajectories τ_{i1} and τ_{i2}, and receiving a preference y_i between them; and (c) updating the posterior (FEEDBACK – Algorithm 3). In contrast to conventional posterior sampling with absolute feedback, DPS samples two policies rather than one at each iteration and solves a credit assignment problem to learn from feedback.

ADVANCE (Algorithm 2) samples from the Bayesian posteriors of the dynamics and utility models to select a policy to roll out. The sampled dynamics and utilities form an MDP, for which value iteration derives the optimal policy π under the sample. One can also view π as a random function whose randomness depends on the sampling of the dynamics and utility models. In the Bayesian setting, it can be shown that π is sampled according to its posterior probability of being the optimal policy π^*. 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, implying 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 that the dynamics are fully-observed; we model the dynamics prior via a Dirichlet distribution for each state-action pair, with conjugate multinomial observation likelihoods. In contrast, performing Bayesian inference over state-action utilities from trajectory-level feedback is much more challenging. We consider a range of approaches (see Appendix B), and found Bayesian linear regression (Section 4.1) to both perform well and admit tractable analysis within our theoretical framework.

4.1 BAYESIAN LINEAR REGRESSION FOR UTILITY INFERENCE AND CREDIT ASSIGNMENT

Credit assignment is the problem of inferring which state-action pairs are responsible for observed trajectory-level preferences. We detail a Bayesian linear regression approach to addressing this task in our setting.

Algorithm 2 ADVANCE: Sample policy from dynamics and utility models

Input: f_p, f_r
Sample $\tilde{\boldsymbol{p}} \sim f_p(\cdot)$ {Sample MDP transition dynamics parameters from posterior}
Sample $\tilde{\boldsymbol{r}} \sim f_r(\cdot)$ {Sample utilities from posterior}
Compute $\pi = \text{argmax}_\pi V(\tilde{\boldsymbol{p}}, \tilde{\boldsymbol{r}})$ {Value iteration yields sampled MDP's optimal policy}
Return π

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

Input: history $\mathcal{H}, f_p, f_r$
Apply Bayesian update to f_p, given $\mathcal{H}$ {Update dynamics model given history}
Apply Bayesian update to f_r, given $\mathcal{H}$ {Update utility model given preferences}
Return f_p, f_r

Let n be the number of iterations, or trajectory pairs, observed so far. Then, the maximum a posteriori (MAP) estimate of the rewards $\overline{\boldsymbol{r}}$ is calculated via ridge regression, similarly to algorithms for the linear bandit setting:

$$\hat{\boldsymbol{r}}_n = M_n^{-1} \sum_{i=1}^{n-1} y_i \boldsymbol{x}_i, \text{ where} \tag{3}$$

$$M_n = \lambda I + \sum_{i=1}^{n-1} \boldsymbol{x}_i \boldsymbol{x}_i^T, \text{ and } \lambda \geq 1. \tag{4}$$

We perform Thompson sampling as in Agrawal and Goyal (2013) and Abeille and Lazaric (2017), such that in iteration n, rewards are sampled from the distribution:

$$\tilde{\boldsymbol{r}}_{n1}, \tilde{\boldsymbol{r}}_{n2} \sim \mathcal{N}(\hat{\boldsymbol{r}}_n, \beta_n(\delta)^2 M_n^{-1}), \text{ where} \tag{5}$$

$$\beta_n(\delta) = R\sqrt{2\log\left(\frac{\det(M_n)^{1/2}\lambda^{-d/2}}{\delta}\right)} + \sqrt{\lambda}S_r$$

$$\leq R\sqrt{d\log\left(\frac{1+\frac{L^2 n}{d\lambda}}{\delta}\right)} + \sqrt{\lambda}S_r,$$

and where $\delta \in (0,1)$ is a failure probability and for all n, $||\boldsymbol{x}_n||_2 \leq L$. Note that $L \leq 2h$, since $||\boldsymbol{x}_n||_2 = ||\boldsymbol{x}_{n2} - \boldsymbol{x}_{n1}||_2 \leq ||\boldsymbol{x}_{n2} - \boldsymbol{x}_{n1}||_1 \leq ||\boldsymbol{x}_{n2}||_1 + ||\boldsymbol{x}_{n1}||_1 = 2h$.

The factor $\beta_n(\delta)$, introduced in Abbasi-Yadkori et al. (2011), is critical to deriving the theoretical guarantees for posterior sampling with linear bandits in Agrawal and Goyal (2013) and Abeille and Lazaric (2017), due to their dependence on Theorems 1 and 2 of Abbasi-Yadkori et al. (2011). Our analysis invokes these results as well. Both of the theorems require any noise in the labels y_n to be sub-Gaussian; in our case, sub-Gaussianity

holds by Assumption 3, as we adopted the linear preference noise model with link function g_{lin}.

Our theoretical analysis is quite different from that for linear bandits in Agrawal and Goyal (2013) and Abeille and Lazaric (2017), because in our setting, observations $\boldsymbol{x}_n$ are *differences* of trajectory feature vectors, policies are chosen via value iteration, and trajectories are obtained by rolling out RL policies while subject to the environment's state transition dynamics.

5 THEORETICAL RESULTS

This section sketches our analysis of the asymptotic Bayesian regret of DPS under a Bayesian linear regression credit assignment model. Appendix A details the full proof, while Appendix B.4 discusses possible future extensions to additional credit assignment models.

The analysis follows three main steps: 1) we prove that DPS is asymptotically-consistent, that is, the probability with which DPS selects the optimal policy approaches 1 over time (Appendix A.1); 2) we asymptotically bound the one-sided Bayesian regret for π_{i2} under the setting where, at each iteration i, DPS only selects policy π_{i2}, while policy π_{i1} is sampled from a fixed distribution over policies (Appendix A.2); and lastly, 3) we assume DPS selects policy π_{i2}, while the π_{i1}-distribution is drifting but converging, and then we asymptotically bound the one-sided regret for π_{i2} (Appendix A.3). Due to the asymptotic consistency shown in 1), the policies are indeed sampled from converging distributions, and so the asymptotic regret rate in 3) holds.

This outline is inspired by the analysis for Self-Sparring (Sui et al., 2017); however, because their guarantee is for dueling bandits with independent Beta-Bernoulli reward models for each action, the details of our analysis are completely different from theirs. Below, we give intuition for each of the three portions of the proof.

Asymptotic consistency of DPS. To prove that DPS is asymptotically consistent, we first prove that samples of the dynamics and reward parameters converge in distribution to their true values:

Proposition 1. *The sampled dynamics converge in distribution to their true values as the* DPS *iteration increases.*

Proof sketch. Applying standard concentration inequalities to the Dirichlet dynamics posterior, one can show that the sampled dynamics converge in distribution to their true values if every state-action pair is visited infinitely-often. The latter condition can be proven via contradiction: assuming that certain state-action pairs are visited finitely-often, DPS does not receive new information about their rewards. Examining their reward posteriors, we show that DPS is guaranteed to eventually sample high enough rewards in the unvisited state-actions that its policies will attempt to reach them.

We also show that with high probability, the sampled rewards exhibit aymptotic consistency:

Proposition 2. *With probability $1-\delta$, where δ is a parameter of the Bayesian linear regression model, the sampled rewards converge in distribution to the true reward parameters, $\overline{\boldsymbol{r}}$, as the* DPS *iteration increases.*

Proof sketch. We leverage Theorem 2 from Abbasi-Yadkori et al. (2011) (Lemma 4 in Appendix A.4): under stated conditions and for any $\delta > 0$, with probability $1-\delta$ and for all $i > 0$, $||\hat{\boldsymbol{r}}_i - \overline{\boldsymbol{r}}||_{M_i} \leq \beta_i(\delta)$. This result defines a high-confidence ellipsoid, which can be linked to the posterior sampling distribution. We demonstrate that it suffices to show that all eigenvalues of the posterior covariance matrix, $\beta_i(\delta)^2 M_i^{-1}$, converge in distribution to zero. This statement is proven via contradiction: we analyze the behavior of posterior sampling if this does not hold. The probability of failure δ comes entirely from Theorem 2 in Abbasi-Yadkori et al. (2011).

From the asymptotic consistency of the dynamics and reward samples, it is straightforward to show that the sampled policies converge to the optimal policy:

Theorem 1. *With probability $1-\delta$, the sampled policies π_{i1}, π_{i2} converge in distribution to the optimal policy, π^*, as $i \longrightarrow \infty$. That is, $P(\pi_{i1} = \pi^*) \longrightarrow 1$ and $P(\pi_{i2} = \pi^*) \longrightarrow 1$ as $i \longrightarrow \infty$.*

Bounding the one-sided regret under a fixed π_{i1}-distribution. To analyze the Bayesian regret of DPS, we adapt the information-theoretic posterior sampling analysis in Russo and Van Roy (2016) to the PBRL setting. In comparison to Russo and Van Roy's work, this requires accounting for preference feedback and incorporating state transition dynamics. Their analysis hinges upon defining a quantity called the *information ratio*, which captures the trade-off between exploration and exploitation. In our setting, we define the information ratio corresponding to the one-sided regret of π_{i2} as:

$$\Gamma_i := \frac{\mathbb{E}_i[y_i^* - y_i]^2}{I_i(\pi^*; (\pi_{i2}, \tau_{i1}, \tau_{i2}, \boldsymbol{x}_{i2} - \boldsymbol{x}_{i1}, y_i))},$$

where y_i is the label in iteration i, y_i^* is the label in iteration i given $\pi_{i2} = \pi^*$, $I(\cdot;\cdot)$ denotes mutual information, and the subscripts i in $\mathbb{E}_i[\cdot]$ and $I_i(\cdot;\cdot)$ indicate conditioning upon the history, as formalized in Appendix A.2. The ratio Γ_i is between the squared instantaneous one-sided regret of π_{i2} (exploitation) and the information gained about the optimal policy (exploration).

When π_{i1} is drawn from a fixed distribution, we show that analogously to Russo and Van Roy (2016), the Bayesian one-sided regret $\mathbb{E}[\text{REG}_2(T)]$ for π_{i2} can be bounded in terms of an upper bound on Γ_i:

Lemma 12. *If $\Gamma_i \leq \overline{\Gamma}$ almost surely for each $i \in \{1, \ldots, N\}$, where N is the number of* DPS *iterations (over which the policies π_{i2} take $T = Nh$ actions), then:*

$$\mathbb{E}[\text{REG}_2(T)] = \mathbb{E}[\text{REG}_2(Nh)] \leq \sqrt{\overline{\Gamma} H(\pi^*) N},$$

where $H(\pi^*)$ is the entropy of the optimal policy π^*. Because there are at most A^{Sh} deterministic policies, $H(\pi^*) \leq \log |A^{Sh}| = Sh \log A$. Substituting this,

$$\mathbb{E}[\text{REG}_2(T)] \leq \sqrt{\overline{\Gamma} ShN \log A} = \sqrt{\overline{\Gamma} ST \log A}.$$

We show that Γ_i can be asymptotically upper-bounded such that $\lim_{i \longrightarrow \infty} \Gamma_i \leq \frac{SA}{2}$, and consequently:

Theorem 2. *If the policy π_{i1} is drawn from a fixed distribution for all i, then for the competing policy π_{i2},* DPS *achieves a one-sided asymptotic Bayesian regret rate of $S\sqrt{\frac{AT \log A}{2}}$.*

The bounds in Lemma 12 and Theorem 2 are asymptotic rather than finite-time, due to the convergence in distribution of the dynamics. If the dynamics are known a priori, then these would be finite-time guarantees; in fact, to prove Lemma 12, we first show that under known dynamics, $\Gamma_i \leq \frac{SA}{2}$ for all i, and then extend the analysis to prove that under converging dynamics, the result still holds asymptotically. Note that in the PBRL setting, it is significantly more difficult to learn the rewards via credit assignment than to learn the dynamics, which are assumed to be fully-observed. Thus, in practice, we expect that DPS would learn the dynamics much faster than the rewards, and so it is reasonable to consider convergence of the dynamics model only asymptotically.

Bounding the one-sided regret under a converging π_{i1}-distribution. Finally, we assume that the distribution of π_{i1} is no longer fixed, but rather converges to some fixed distribution over deterministic policies. To asymptotically bound the one-sided regret incurred by π_{i2}, we leverage that when two discrete random variables converge in distribution, such that $X_n \xrightarrow{D} X$ and $Y_n \xrightarrow{D} Y$, their mutual information also converges: $\lim_{n \longrightarrow \infty} I(X_n, Y_n) = I(X, Y)$. This fact allows us to bound the one-sided regret for π_{i2} as follows:

Lemma 17. *Assume that the sampling distribution of π_{i1} converges to a fixed probability distribution. Then, the information ratio Γ_i corresponding to π_{i2}'s one-sided regret $\mathbb{E}[\text{REG}_2(T)]$ satisfies $\lim_{i \longrightarrow \infty} \Gamma_i \leq \frac{SA}{2}$.*

Combining Lemma 17 with the asymptotic consistency of sampled policies as shown in Theorem 1, $P(\pi_{i1} = \pi^*) \longrightarrow 1$, yields our main theoretical result:

Theorem 3. *With probability $1-\delta$, where δ is a parameter of the Bayesian linear regression model, the expected Bayesian regret $\mathbb{E}[\text{REG}(T)]$ of* DPS *achieves an asymptotic rate of $S\sqrt{2AT \log A}$.*

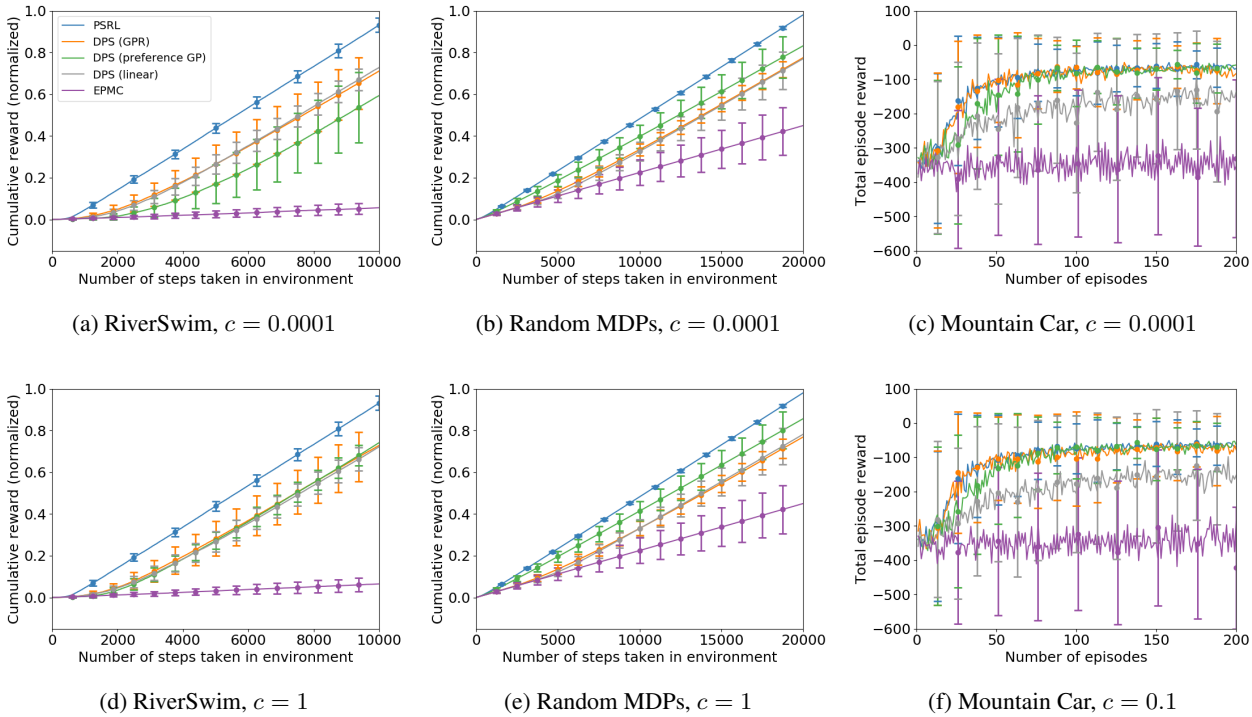
Discussion. The specific theoretical results presented yield a high-probability asymptotic Bayesian no-regret rate for DPS under Bayesian linear regression credit assignment. The proof consists of first demonstrating that the algorithm is asymptotically consistent, and then analyzing its information ratio to characterize the Bayesian regret. We adopted this information-theoretic perspective because we found it more amenable to preference-based feedback than other prevalent methods from the linear bandits literature.

In particular, while several existing regret analyses for posterior sampling with linear bandits (Agrawal and Goyal, 2013; Abeille and Lazaric, 2017) are based upon martingale concentration properties derived in Abbasi-Yadkori et al. (2011), we found that these techniques cannot readily extend to the preference-feedback setting (Appendix C). These linear bandit analyses assume that each observation $\boldsymbol{x}_i$ that incurs regret contributes fully toward learning the rewards. In contrast, we assume that while regret is incurred with respect to the observations $\boldsymbol{x}_{i1}, \boldsymbol{x}_{i2}$, learning occurs only with respect to observation *differences*, $\boldsymbol{x}_i = \boldsymbol{x}_{i2} - \boldsymbol{x}_{i1}$. In preference-based learning settings, it is common to make such assumptions as $P(\tau_{i2} \succ \tau_{i1}) = g(\boldsymbol{x}_{i2} - \boldsymbol{x}_{i1})$, for some function g. In comparison to the martingale-based techniques, the information ratio provides a more direct method for quantifying the trade-off between exploration and exploitation.

Theoretically analyzing other credit assignment models, in addition to Bayesian linear regression, is an important direction for future work. We conjecture that our proof methodology could extend toward other asymptotically-consistent credit assignment models. Indeed, recent work (Dong and Van Roy, 2018) has analyzed the information ratio for more general link functions, including for logistic bandits. It would be interesting to study the information ratio's behavior under general link functions, as well as to characterize its relationship to the dynamics model's convergence. It would also be interesting to develop methodology for extending the analysis to achieve finite-time convergence guarantees.

6 EXPERIMENTS

We validate the empirical performance of DPS in three simulated domains with varying degrees of preference



(a) RiverSwim, $c = 0.0001$ (b) Random MDPs, $c = 0.0001$ (c) Mountain Car, $c = 0.0001$

(d) RiverSwim, $c = 1$ (e) Random MDPs, $c = 1$ (f) Mountain Car, $c = 0.1$

Figure 1: Empirical performance of DPS; each simulated environment is shown under the two least-noisy user preference models evaluated. The plots show DPS with three credit assignment models: Gaussian process regression (GPR), Bayesian linear regression, and a Gaussian process preference model. PSRL is an upper bound that receives numerical rewards, while EPMC is a baseline. Plots display the mean +/- one standard deviation over 100 runs of each algorithm tested. The remaining user noise models are plotted in Appendix D. For RiverSwim and Random MDPs, normalization is with respect to the total reward achieved by the optimal policy. Overall, we see that DPS performs well and is robust to the choice of credit assignment model.

noise and using three alternative credit assignment models. We find that DPS generally performs well and compares favorably against standard PBRL baselines.

Experimental setup. We evaluate on three simulated environments: RiverSwim and random MDPs (described in Osband et al. (2013)) and the Mountain Car problem as detailed in Wirth (2017). The RiverSwim environment has six states and two actions (actions 0 and 1); the optimal policy always chooses action 1, which maximizes the probability of reaching a goal state-action pair. Meanwhile, a suboptimal policy—yielding a small 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 environment, we generate random MDPs with 10 states and 5 actions. The transition dynamics and rewards are respectively generated from Dirichlet (all parameters set to 0.1) and exponential (rate parameter $=$ 5) distributions. These distribution parameters were chosen to generate MDPs with sparse dynamics and rewards. For each random MDP, the sampled reward values were shifted and normalized so that the minimum reward is zero and their mean is one.

Thirdly, in the Mountain Car problem, an under-powered car in a valley must reach the top of a hill by accelerating in both directions to build its momentum. The state space is two-dimensional (position and velocity), while there are three actions (left, right, and neutral). Our implementation begins each episode in a uniformly-random state and has a maximum episode length of 500. We discretize the state space into 10 states in each dimension. Each episode terminates either when the car reaches the goal or after 500 steps, and rewards are -1 in every step.

In each environment, preferences between trajectory pairs were generated by (noisily) comparing their total accrued rewards; this reward information was hidden from the learning algorithm, which observed only the trajectory preferences and state transitions. For trajectories τ_i and τ_j with total rewards $\overline{r}(\tau_i)$ and $\overline{r}(\tau_j)$, we con-

sider two models for generating preferences: a) a logistic model, $P(\tau_i \succ \tau_j) = \{1 + \exp[-(\overline{r}(\tau_i) - \overline{r}(\tau_j))/c]\}^{-1}$, and b) a linear model, $P(\tau_i \succ \tau_j) = (\overline{r}(\tau_i) - \overline{r}(\tau_j))/c$, where in both cases, the temperature c controls the degree of noisiness. In the linear case, c is assumed to be large enough that $P(\tau_i \succ \tau_j) \in [0, 1]$. Note that in ties where $\overline{r}(\tau_i) = \overline{r}(\tau_j)$, preferences are uniformly-random.

Methods compared. We evaluate DPS under three credit assignment models (Appendix B): 1) Bayesian linear regression, 2) Gaussian process regression, and 3) a Gaussian process preference model. User noise generated via the logistic model has noise levels: $c \in \{10, 2, 1, 0.001\}$ for RiverSwim and random MDPs and $c \in \{100, 20, 10, 0.001\}$ for the Mountain Car. We selected higher values of c for the Mountain Car because $|\overline{r}(\tau_i) - \overline{r}(\tau_j)|$ has a wider range. Additionally, we evaluate the linear preference noise model with $c = 2h\Delta\overline{\boldsymbol{r}}$, where $\Delta\overline{\boldsymbol{r}}$ is the difference between the maximum and minimum element of $\overline{\boldsymbol{r}}$ for each MDP; this choice of c guarantees that $P(\tau_i \succ \tau_j) \in [0, 1]$, but yields noisier preferences than the logistic noise models considered.

As discussed in Section 2, many existing PBRL algorithms handle a somewhat distinct setting from ours, as they assume access to a simulator between preference queries and/or prioritize minimizing preference queries rather than online regret. As a baseline, we evaluate the Every-Visit Preference Monte Carlo (EPMC) algorithm with probabilistic credit assignment (Wirth and Fürnkranz, 2013b; Wirth, 2017). While EPMC does not require simulations between preference queries, it has several limitations, including: 1) the exploration approach always takes uniformly-random actions with some probability, and thus, the authors' plots do not depict online reward accumulation, and 2) EPMC assumes that compared trajectories start in the same state. Lastly, we compare against the posterior sampling RL algorithm (PSRL) from Osband et al. (2013), which receives the true numerical rewards at each step, and thus upper-bounds the achievable performance of a preference-based algorithm.

Results. Figure 1 depicts performance curves for the three environments, each with two noise models (Appendix D contains additional results and details). DPS performs well in all simulations, and significantly outperforms the EPMC baseline. In RiverSwim, most credit assignment models perform best in the second-to-least-noisy case (logistic noise, $c = 1$), since it is harder to escape the local minimum under the least-noisy preferences. We also see that DPS is competitive with PSRL, which has access to the full cardinal rewards at each state-action. Additionally, while our theoretical guarantees for DPS assume fixed-horizon episodes, the Mountain Car results demonstrate that it also succeeds with variable episode lengths. Finally, the performance of DPS is robust to the choice of credit assignment model, and in fact using Gaussian processes (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 models as subroutines.

7 CONCLUSION

This work investigates the preference-based reinforcement learning problem, in which an RL agent receives comparative preferences instead of absolute real-valued rewards as feedback. We develop the DUELING POSTERIOR SAMPLING (DPS) algorithm, which optimizes policies in a 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, making it both a theoretically-justified and practically-promising algorithm.

There are many directions for future work. Assumptions governing the user's preferences, such as requiring an underlying utility model, could be relaxed. It would also be interesting to extend our theoretical analysis to additional credit assignment approaches and to pursue finite-time guarantees. We expect that DPS would perform well with any asymptotically-consistent reward model that sufficiently captures users' preference behavior, and hope to develop models that are tractable with larger state and action spaces. For instance, incorporating kernelized input spaces could further improve sample efficiency.

Acknowledgments

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

References

Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári. Improved algorithms for linear stochastic bandits. In *NeurIPS*, 2011.

M. Abeille and A. Lazaric. Linear Thompson sampling revisited. *Electronic Journal of Statistics*, 11(2), 2017.

S. Agrawal and N. Goyal. Thompson sampling for contextual bandits with linear payoffs. In *ICML*, 2013.

S. Agrawal and R. Jia. Optimistic posterior sampling for reinforcement learning: Worst-case regret bounds. In *NeurIPS*, 2017.

N. Ailon, Z. Karnin, and T. Joachims. Reducing dueling bandits to cardinal bandits. In *ICML*, 2014.

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

R. Akrour, M. Schoenauer, M. Sebag, and J.-C. Souplet. Programming by feedback. In *ICML*, volume 32, 2014.

D. Amodei, C. Olah, J. Steinhardt, P. Christiano, et al. Concrete problems in AI safety. *arXiv preprint*, 2016.

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

C. Basu, Q. Yang, D. Hungerman, M. Sinahal, and A. D. Dragan. Do you want your autonomous car to drive like you? In *Int. Conf. on Human-Robot Interaction*. IEEE, 2017.

C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, et al. Learning to rank using gradient descent. In *ICML*, 2005.

C. J. Burges, R. Ragno, and Q. V. Le. Learning to rank with nonsmooth cost functions. In *NeurIPS*, 2007.

R. Busa-Fekete, B. Szörényi, P. Weng, W. Cheng, and E. Hüllermeier. Preference-based evolutionary direct policy search. In *ICRA Workshop on Autonomous Learning*, 2013.

O. Chapelle and L. Li. An empirical evaluation of Thompson sampling. In *NeurIPS*, 2011.

P. F. Christiano, J. Leike, T. Brown, et al. Deep reinforcement learning from human preferences. In *NeurIPS*, 2017.

W. Chu and Z. Ghahramani. Preference learning with Gaussian processes. In *ICML*, 2005.

S. Dong and B. Van Roy. An information-theoretic analysis for Thompson sampling with many actions. In *NeurIPS*, 2018.

M. Dudík, K. Hofmann, R. E. Schapire, A. Slivkins, and M. Zoghi. Contextual dueling bandits. In *COLT*, 2015.

B. Eric, N. D. Freitas, and A. Ghosh. Active preference learning with discrete choice data. In *NeurIPS*, 2008.

J. Fürnkranz, E. Hüllermeier, W. Cheng, et al. Preference-based reinforcement learning: A formal framework and a policy iteration algorithm. *Machine learning*, 89(1-2), 2012.

A. Gopalan and S. Mannor. Thompson sampling for learning parameterized Markov decision processes. In *COLT*, 2015.

R. Herbrich, T. Graepel, and K. Obermayer. Support vector learning for ordinal regression. In *International Conference on Artificial Neural Networks*, volume 1. IET, 1999.

N. Houlsby, F. Huszár, et al. Bayesian active learning for classification and preference learning. *arXiv*, 2011.

T. Jaksch, R. Ortner, and P. Auer. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11, 2010.

T. Joachims. A support vector method for multivariate performance measures. In *ICML*. ACM, 2005.

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

T.-Y. Liu. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval*, 3(3), 2009.

N. Nikolov, J. Kirschner, et al. Information-directed exploration for deep reinforcement learning. *arXiv*, 2018.

I. Osband and B. Van Roy. Why is posterior sampling better than optimism for reinforcement learning? In *ICML*, 2017.

I. Osband, D. Russo, et al. (More) efficient reinforcement learning via posterior sampling. In *NeurIPS*, 2013.

F. Radlinski and T. Joachims. Query chains: Learning to rank from implicit feedback. In *SIGKDD*. ACM, 2005.

S. Y. Ramamohan, A. Rajkumar, and S. Agarwal. Dueling bandits: Beyond Condorcet winners to general tournament solutions. In *NeurIPS*, 2016.

K. Raman, T. Joachims, P. Shivaswamy, and T. Schnabel. Stable coactive learning via perturbation. In *ICML*, 2013.

D. Russo and B. Van Roy. An information-theoretic analysis of Thompson sampling. *The Journal of Machine Learning Research*, 17(1), 2016.

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

P. Shivaswamy and T. Joachims. Online structured prediction via coactive learning. In *ICML*, 2012.

P. Shivaswamy and T. Joachims. Coactive learning. *Journal of Artificial Intelligence Research*, 53, 2015.

Y. Sui, V. Zhuang, J. W. Burdick, and Y. Yue. Multi-dueling bandits with dependent arms. In *UAI*, 2017.

Y. Sui, V. Zhuang, J. Burdick, et al. Stagewise safe Bayesian optimization with Gaussian processes. In *ICML*, 2018a.

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

B. Szörényi, R. Busa-Fekete, A. Paul, and E. Hüllermeier. Online rank elicitation for Plackett-Luce: A dueling bandits approach. In *NeurIPS*, 2015.

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

A. Wilson, A. Fern, et al. A Bayesian approach for policy learning from trajectory preference queries. In *NeurIPS*, 2012.

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

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

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

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

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

H. Wu and X. Liu. Double Thompson sampling for dueling bandits. In *NeurIPS*, 2016.

Y. Yue, T. Finley, et al. A support vector method for optimizing average precision. In *Int. SIGIR conf. on research and development in information retrieval*. ACM, 2007.

Y. Yue, J. Broder, et al. The k-armed dueling bandits problem. *Journal of Computer and System Sciences*, 78(5), 2012.

A. Zanette and R. Sarkar. Information directed reinforcement learning. Technical report, Technical report, 2017.

M. Zoghi, S. Whiteson, et al. Relative upper confidence bound for the k-armed dueling bandit problem. In *ICML*, 2014.

M. Zoghi, Z. S. Karnin, S. Whiteson, and M. De Rijke. Copeland dueling bandits. In *NeurIPS*, 2015.