Preference-based reinforcement learning: evolutionary direct policy search using a preference-based racing algorithm

Róbert Busa-Fekete, Balázs Szörényi, Paul Weng, Weiwei Cheng, Eyke Hüllermeier

Preference-based Evolutionary Direct Policy Search

R. Busa-Fekete W. Cheng E. Hüllermeier *
B. Szörenyi † P. Weng ‡

Abstract

We introduce a novel approach to preference-based reinforcement learning, namely a preference-based variant of a direct policy search method based on evolutionary optimization. The core of our approach is a preference-based racing algorithm that selects the best among a given set of candidate policies with high probability. To this end, the algorithm operates on a suitable ordinal preference structure and only uses pairwise comparisons between sample rollouts of the policies. Embedding the racing algorithm in a rank-based evolutionary search procedure, we show that approximations of the so-called Smith set of optimal policies can be produced with certain theoretical guarantees. Apart from a formal performance and complexity analysis, we present first experimental studies showing that our approach performs well in practice.

1 Introduction

Preference-based reinforcement learning (PBRL) is a novel research direction combining reinforcement learning (RL) and preference learning [11]. It aims at extending existing RL methods so as to make them amenable to training information and external feedback more general than numerical rewards, which are often difficult to obtain or expensive to compute. For example, anticipating our experimental study in the domain of medical treatment planning, to which we shall return in Section 5, how to specify the cost of a patient's death in terms of a reasonable numerical value?

In [2] and [7], the authors tackle the problem of learning policies solely on the basis of qualitative preference information, namely pairwise comparisons

*Computational Intelligence Group, Department of Mathematics and Computer Science, University of Marburg, Germany {busarobi,cheng,eyke}@mathematik.uni-marburg.de

†INRIA Lille - Nord Europe, SequeL project, 40 avenue Halley, 59650 Villeneuve d'Ascq, France szorenyi@inf.u-szeged.hu

‡Laboratory of Computer Science of Paris 6, University Pierre and Marie Curie, 4 place Jussieu, 75005 Paris, France paul.weng@lip6.fr

between trajectories; such comparisons suggest that one system behavior is preferred to another one, but without committing to precise numerical rewards. Building on novel methods for preference learning, this is accomplished by providing the RL agent with qualitative policy models, such as ranking functions. More specifically, Cheng et al. [7] use a method called *label ranking* to train a model that ranks actions given a state; their approach generalizes classification-based approximate policy iteration [19]. Instead of ranking actions given states, Akrour at al. [2] exploit preferences on trajectories in order to learn a model that ranks complete policies.

In this paper, we present a preference-based extension of *evolutionary direct policy search* (EDPS) as proposed by Heidrich-Meisner and Igel [15]. As a direct policy search method, it shares commonalities with [2], but also differs in several respects. In particular, their approach (as well as follow-up work of the same authors, such as [3]) is arguably more specialized and tailored for applications in robotics, in which a user interacts with the learner in an iterative process. Moreover, policy search is not performed in a parametrized policy space directly but in a *feature space* capturing important background knowledge about the task to be solved.

EDPS casts policy learning as a search problem in a parametric policy space, where the function to be optimized is a performance measure like expected total reward, and evolution strategies (ES) such as CMA-ES [13] are used as optimizers. Moreover, since the evaluation of a policy can only be done approximately, namely in terms of a finite number of *rollouts*, the authors make use of *racing algorithms* to control this number in an adaptive manner. These algorithms return a sufficiently reliable ranking over the current set of policies (candidate solutions), which is then used by the ES for updating its parameters and population. A key idea of our approach is to extend EDPS by replacing the *value-based* racing algorithm with a *preference-based* one. Correspondingly, the development of a preference-based racing algorithm can be seen as a core contribution of this paper.

In the next section, we recall the original RL setting and the EDPS framework for policy learning. Our preference-based generalization of this framework is introduced in Section 3. A key component of our approach, the preference-based racing algorithm, is detailed and analyzed in Section 4. Experiments are presented in Section 5. Section 6 provides an overview of related work and Section 7 concludes the paper.

2 Evolutionary Direct Policy Search

We start by introducing notation to be used throughout the paper. A Markov Decision Process (MDP) is a 4-tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathbf{P}, r)$, where $\mathcal{S}$ is the (possibly infinite) state space and $\mathcal{A}$ the (possibly infinite) set of actions.

$$\mathbf{P} : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \to [0, 1]$$

is the transition probability that defines the random transitions $\mathbf{s}' \sim \mathbf{P}(\cdot \,|\, \mathbf{s}, a)$ from a state $\mathbf{s}$ applying action a, and $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the *reward function*, i.e., $r(\mathbf{s}, a)$ defines the reward for taking action $a \in \mathcal{A}$ in state $\mathbf{s} \in \mathcal{S}$.

We will only consider *undiscounted* and *episodic* MDPs with a finite horizon $T \in \mathbb{N}^+$. In the episodic setup, there is a set of *initial* states $\mathcal{S}_0 \subseteq \mathcal{S}$ and a set of *terminal* states $\mathcal{S}_\infty \subseteq \mathcal{S}$ which are impossible to leave (and in which no reward is incurred). A *policy* $\pi : \mathcal{S} \to \mathcal{A}$ assigns an action to each state.

$\mathcal{H}^{(T)} = (\mathcal{S} \times \mathcal{A})^T \times S_\infty$ is the set of *admissible histories* with time horizon T. A *finite history* or simply *history* is a state/action sequence

$$\mathbf{h} = \left(\mathbf{s}^{(1)}, a^{(1)}, \ldots, a^{(T)}, \mathbf{s}^{(T+1)}\right) \in \mathcal{H}^{(T)}$$

that starts from an initial state $\mathbf{s}^{(1)} \in \mathcal{S}_0$ drawn from a user-defined *initial state distribution* $\mathbf{P}_0$ over $\mathcal{S}_0$. Each history $\mathbf{h}$ uniquely determines a sequence of rewards, so a *return function* $V : \mathcal{H}^{(T)} \to \mathbb{R}$ can be defined as

$$V(\mathbf{h}) = \sum_{i=1}^{T} r(\mathbf{s}^{(i)}, a^{(i)}) \ .$$

We write $\mathbf{h}_\pi$ for a history generated by following policy π, that is, $\pi(\mathbf{s}^{(t)}) = a^{(t)}$ for all $t \in [T] = \{1, \ldots, T\}$.

2.1 The EDPS framework

We briefly outline the *evolutionary direct policy search* (EDPS) approach introduced by Heidrich-Meisner and Igel [15]. Assume a parametric policy space

$$\Pi = \{\pi_\Theta \,|\, \Theta \in \mathbb{R}^p\} \ ,$$

i.e., a space of policies parametrized by a vector Θ. For example, if $\mathcal{S} \subseteq \mathbb{R}^p$, this could simply be a class of linear policies $\pi_\Theta(\mathbf{s}) = \Theta^T \mathbf{s}$. Searching a good policy can be seen as an optimization problem where the search space is the parameter space and the target function is a policy performance evaluation, such as expected total reward.

This optimization-based policy search framework, which is called *direct policy search*, has two main branches: *gradient-based* and *gradient-free* methods. Gradient-based methods like the REINFORCE algorithm [30] estimate the gradient of the policy parameters to guide the optimizer. Gradient-free methods, on the other hand, make use of a *black-box optimizer* such as *evolution strategies* [6], which gave rise to the EDPS approach.

2.2 Evolutionary optimization

Evolution strategies (ES) are population-based, randomized search techniques that maintain a set of candidate solutions $\Theta_1, \ldots, \Theta_\mu$ (the population) and a set of (auxiliary) parameters Ω over the search space. An ES optimizer is an iterative method that repeats the following steps in each iteration t:

(i) sample a set of λ candidate solutions $\{\Theta_j^{(t+1)}\}_{j=1}^{\lambda}$, called *offspring population*, from the current model defined by $\Omega^{(t)}$ and the *parent population* $\{\Theta_i^{(t)}\}_{i=1}^{\mu}$;

(ii) evaluate each offspring solution and select the best μ ones as a new parent population;

(iii) update $\Omega^{(t)}$ based on the new parent population.

The use of evolution strategies proved to be efficient in direct policy search [14]. In the EDPS method by Heidrich-Meisner and Igel [15], an ES is applied for optimizing the *expected total reward* over the parameter space of linear policies. To this end, the expected total reward of a policy is estimated based on a so-called *rollout set*. More specifically, for an MDP $\mathcal{M}$ with initial distribution $\mathbf{P}_0$, each policy π generates a probability distribution $\mathbf{P}_\pi$ over the set of histories $\mathcal{H}^{(T)}$. Then, the *expected total reward* of π can be written as $\rho_\pi = \mathbb{E}_{\mathbf{h}\sim\mathbf{P}_\pi}[V(\mathbf{h})]$ [25], and the expectation according to $\mathbf{P}_\pi$ can be estimated by the average return over a rollout set $\{\mathbf{h}_\pi^{(i)}\}_{i=1}^{n}$.

From a practical point of view, the size of the rollout set is crucial: On the one hand, the learning process gets slow if n is large, while on the other hand, the ranking over the offspring population is not reliable enough if the number of rollouts is too small; in that case, there is a danger of selecting a suboptimal subset of the offspring population instead of the best μ ones. Therefore, [15] proposed to apply an adaptive uncertainty handling scheme, called *racing algorithm*, for controlling the size of rollout sets in a optimal way.

Their EDPS framework is described schematically in Algorithm 1. It bears a close resemblance to ES, but the selection step (line 7) is augmented with a racing algorithm that generates histories for each of the current policies $\pi_{\Theta_i^{(t)}}$ by sampling from the corresponding distribution in an adaptive manner until being able to select the best μ policies based on their expected total reward estimates with probability at least $1-\delta$ (see Section 2.3). The parameter $n_{\max}$ specifies an upper bound on the number of rollouts for a single policy. The racing algorithm returns a ranking over the policies in the form of a permutation σ.

2.3 Value-based racing

Generating a history in an MDP by following policy π is equivalent to drawing an example from $\mathbf{P}_\pi$. Consequently, a policy along with an MDP and initial distribution can simply be seen as a random variable. Therefore, to make our presentation of the racing algorithm more general, we shall subsequently consider the problem of comparing random variables.

Let $X_1, \ldots, X_K$ be random variables with respective (unknown) distribution functions $\mathbf{P}_{X_1}, \ldots, \mathbf{P}_{X_K}$. These random variables, subsequently also called *options*, are supposed to have finite expected values $\mu_i = \int x d\mathbf{P}_{X_i}(x)$. The racing task consists of selecting, with a predefined confidence $1-\delta$, a κ-sized subset of the K options with highest expectations. In other words, one seeks

Algorithm 1 EDPS $(\mathcal{M}, \mu, \lambda, n_{\max}, \delta)$

Initialization: select an initial parameter vector $\Omega^{(0)}$ and an initial set of candidate solutions ${\Theta_1}^{(0)}, \ldots, {\Theta_\mu}^{(0)}$, $\sigma^{(0)}$ is the identity permutation
$t = 0$
repeat
  $t = t + 1$
  for $\ell = 1, \ldots, \lambda$ **do** ▷ Sample new solutions
    $\Theta_\ell^{(t)} \sim F(\Omega^{(t-1)}, \Theta^{(t-1)}_{\sigma^{(t-1)}(1)}, \ldots, \Theta^{(t-1)}_{\sigma^{(t-1)}(\mu)})$
  $\sigma^{(t)} = \textbf{Racing}\left(\mathcal{M}, \pi_{\Theta_1^{(t)}}, \ldots, \pi_{\Theta_\lambda^{(t)}}, \mu, n_{\max}, \delta\right)$
  $\Omega^{(t)} = \textbf{Update}(\Omega^{(t-1)}, \Theta^{(t)}_{\sigma^{(t)}(1)}, \ldots, \Theta^{(t)}_{\sigma^{(t)}(\mu)})$
until Stopping criterion fulfilled
return $\pi_{\Theta_1^{(t)}}$

a set $I \subseteq [K]$ of cardinality κ maximizing $\sum_{i \in I} \mu_i$, which is equivalent to the following optimization problem:

$$\sum_{i \in I} \sum_{j \neq i} \mathbb{I}\{\mu_j < \mu_i\} \longrightarrow \max_{I \subseteq [K]:\ |I| = \kappa} , \tag{1}$$

where the indicator function $\mathbb{I}\{\cdot\}$ maps truth degrees to $\{0, 1\}$ in the standard way. This choice problem must be solved on the basis of random samples drawn from $X_1, \ldots, X_K$.

The Hoeffding race (HR) algorithm [21, 22] is an adaptive sampling method that makes use of the Hoeffding bound to construct confidence intervals for the empirical mean estimates of the options. Then, in the case of non-overlapping confidence intervals, some options can be eliminated from further sampling. More precisely, if the upper confidence bound for a particular option is smaller than the lower bound of $K - \kappa$ random variables, then it can be discarded from the solution set I in (1) with high probability; the inclusion of an option in I can be decided analogously (see Figure 1 for an illustration). For a detailed implementation of the HR algorithm, see [15].

3 Preference-based EDPS

The preference-based policy learning settings considered in [12, 2] proceed from a (possibly partial) preference relation $\prec$ over histories $\mathbf{h} \in \mathcal{H}^{(T)}$, and the goal is to find a policy which tends to generate preferred histories with high probability. In this regard, it is notable that, in the EDPS framework, the precise values of the function to be optimized (in this case the expected total rewards) are actually not used by the evolutionary optimizer. Instead, for updating its current state $(\Omega, \Theta_1, \ldots, \Theta_\mu)$, the ES only needs the *ranking* of the candidate solutions. The values are only used by the racing algorithm in order to produce this ranking.

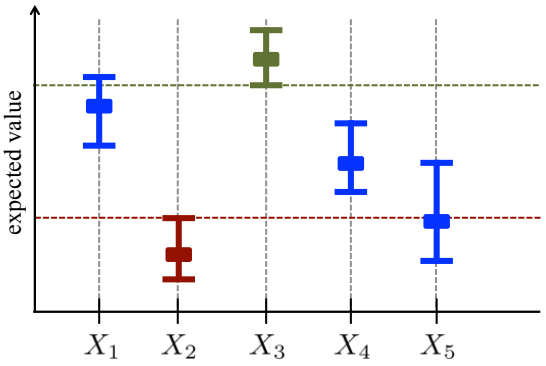


Figure 1: Illustration of the value-based racing problem: The expectations of the random variables are estimated in terms of confidence intervals that shrink in the course of time. In this example, if two options ought to be selected, then X_2 can be discarded, as it is already worse than three other options (with high probability); likewise, the option X_3 will certainly be an element of the top-2 selection, as it has already outperformed three others. For the other options, a decision can not yet be made.

Consequently, an obvious approach to realizing the idea of a purely preference-based version of evolutionary direct policy search (PB-EDPS) is to replace the original racing algorithm (line 7) by a preference-based racing algorithm that only uses pairwise comparisons between policies (or, more specifically, sample histories generated from these policies). We introduce a racing algorithm of this kind in Section 4.

A main prerequisite of such an algorithm is a "lifting" of the preference relation $\prec$ on $\mathcal{H}^{(T)}$ to a preference relation $\ll$ on the space of policies Π; in fact, without a relation of that kind, the problem of ranking policies is not even well-defined. More generally, recalling that we can associate policies with random variables X and histories with realizations $x \in \Omega$, the problem can be posed as follows: Given a (possibly partial) order relation $\prec$ on the set of realizations Ω, how to define a reasonable order relation on the set of probability distributions over Ω which is "learnable" by a preference-based racing algorithm?

A natural definition of the preference relation $\ll$ that we shall adopt in this paper is as follows:

$$X \ll Y \text{ if and only if } \mathbf{P}(Y \prec X) < \mathbf{P}(X \prec Y) \ ,$$

where $\mathbf{P}(Y \prec X)$ denotes the probability that the realization of X is preferred (with respect to $\prec$) to the realization of Y.

Despite the appeal of $\ll$ as an ordinal decision model, this relation is not necessarily transitive and may even have cycles [10]. The preferential structure induced by $\ll$ is well-studied in social choice theory [24], as it is closely related

to the idea of choosing a winner in an election where only pairwise comparisons between candidates are available. We borrow two important notions from social choice theory, namely the *Condorcet winner* and the *Smith set*.

Definition 1. *A random variable X_i is a Condorcet winner among a set of random variables $X_1, \ldots, X_K$ if $X_\ell \ll X_i$ for all $\ell \neq i$.*

Definition 2. *For a set of random variables $\mathcal{X} = \{X_1, \ldots, X_K\}$, the Smith set is the smallest non-empty set $\mathcal{C}^* \subseteq \mathcal{X}$ satisfying $X_j \ll X_i$ for all $X_i \in \mathcal{C}^*$ and $X_j \in \mathcal{X} \setminus \mathcal{C}^*$.*

If the Condorcet winner X^* exists, then it is the greatest element of $\ll$ and $\mathcal{C}^* = \{X^*\}$. More generally, the Smith set $\mathcal{C}^*$ can be interpreted as the smallest non-empty set of options that are "better" than all options outside $\mathcal{C}^*$.

Due to preferential cycles, the (racing) problem of selecting the κ best options may still not be well-defined for $\ll$ as the underlying preference relation. To overcome this difficulty, we refer to the *Copeland relation* $\ll_C$ as a surrogate. For a set $\mathcal{X} = \{X_1, \ldots, X_K\}$ of random variables, it is defined as follows [24]: $X_i \ll_C X_j$ if and only if $d_i < d_j$, where $d_i = \#\{k : X_k \ll X_i, X_k \in \mathcal{X}\}$. Its interpretation is again simple: an option X_i is preferred to X_j whenever X_i "beats" (w.r.t. $\ll$) more options than X_j does. Since the preference relation $\ll_C$ has a numeric representation in terms of the d_i, it is a total preorder. Note that $\ll_C$ is "contextualized" by the set $\mathcal{X}$ of random variables: the comparison of two options X_i and X_j, i.e., whether or not $X_i \ll_C X_j$, also depends on the other alternatives in $\mathcal{X}$.

Obviously, when a Condorcet winner exists, it is the greatest element for $\ll_C$. More generally, the following proposition establishes an important connection between $\ll$ and $\ll_C$.

Proposition 3. *Let $\mathcal{X} = \{X_1, \ldots, X_K\}$ be a set of random variables with Smith set $\mathcal{C}^*$. Then, for any $X_i \in \mathcal{C}^*$ and $X_j \in \mathcal{X} \setminus \mathcal{C}^*$, $X_j \ll_C X_i$.*

Proof. Let $K_{\mathcal{C}^*}$ be the size of $\mathcal{C}^*$. By the definition of the Smith set, $d_i \geq K - K_{\mathcal{C}^*}$ for all $X_i \in \mathcal{C}^*$, since X_i beats all elements of $\mathcal{X} \setminus \mathcal{C}^*$ w.r.t $\ll$. Moreover, $d_j < K - K_{\mathcal{C}^*}$ for all $X_j \in \mathcal{X} \setminus \mathcal{C}^*$, since X_j is beaten by all elements of $\mathcal{C}^*$. Therefore, $d_j < d_i$ for any $X_i \in \mathcal{C}^*$ and $X_j \in \mathcal{X} \setminus \mathcal{C}^*$. □

Therefore, the surrogate relation $\ll_C$ is coherent with the preference order $\ll$ in the sense that the "rational choices", namely the elements of the Smith set, are found on the top of this preorder. In the next section, we shall therefore use $\ll_C$ as an appropriate ordinal decision model for preference-based racing.

4 Preference-based Racing Algorithm

This section is devoted to our preference-based racing algorithm (PBR). Section 4.1 describes the concentration property of the estimate of $\mathbf{P}(X \prec Y)$, which is a cornerstone of our approach. Section 4.2 provides a simple technique to handle incomparability of random samples. Section 4.3 outlines the PBR algorithm as a whole, and Section 4.4 provides a formal analysis of this algorithm.

4.1 An efficient estimator of $\mathbf{P}(X \prec Y)$

In Section 3, we introduced an ordinal decision model specified by the order relation $\ll_C$. Sorting a set of random variables $X_1, \ldots, X_K$ according to $\ll_C$ first of all requires an *efficient estimator* of $S(X_i, X_j) = \mathbf{P}(X_i \prec X_j)$.

A two-sample U-statistic called the *Mann-Whitney U-statistic* (also known as the *Wilcoxon 2-sample statistic*) is an unbiased estimate of $S(\cdot, \cdot)$ [27]. Given independent samples $\mathbf{X} = \{x^{(1)}, \ldots, x^{(n)}\}$ and $\mathbf{Y} = \{y^{(1)}, \ldots, y^{(n)}\}$ of two independent random variables X and Y (for simplicity, we assume equal sample sizes), it is defined as

$$\widehat{S}(\mathbf{X}, \mathbf{Y}) = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \mathbb{I}\{x^{(i)} \prec y^{(j)}\} \ . \tag{2}$$

Apart from being an unbiased estimator of $S(X, Y)$, (2) possesses concentration properties resembling those of the sum of independent random variables.[1]

Theorem 4 ([17], §5b)**.** *For any $\epsilon > 0$, using the notations introduced above,*

$$\mathbf{P}\left(\left|\widehat{S}(\mathbf{X}, \mathbf{Y}) - S(X, Y)\right| \geq \epsilon\right) \leq 2\exp(-2n\epsilon^2) \ .$$

An equivalent formulation of this theorem is as follows: For any $0 < \delta < 1$, the interval

$$\Bigg[\underbrace{\widehat{S}(\mathbf{X}, \mathbf{Y}) - \sqrt{\frac{1}{2n} \ln \frac{2}{\delta}}}_{L(\mathbf{X},\mathbf{Y})}, \underbrace{\widehat{S}(\mathbf{X}, \mathbf{Y}) + \sqrt{\frac{1}{2n} \ln \frac{2}{\delta}}}_{U(\mathbf{X},\mathbf{Y})}\Bigg] \tag{3}$$

contains $S(X, Y)$ with probability at least $1 - \delta$. For more details on the U-statistic, see Appendix A.1.

4.2 Handling incomparability

Recall that $\prec$ is only assumed to be a partial order and, therefore, allows for incomparability $x \perp y$ between realizations x and y of random variables (histories generated by policies). In such cases we have $\mathbb{I}\{x \prec y\} = \mathbb{I}\{y \prec x\} = 0$, and, consequently $\widehat{S}(\mathbf{X}, \mathbf{Y}) + \widehat{S}(\mathbf{Y}, \mathbf{X}) < 1$. Since this inequality is inconvenient and may complicate the implementation of the algorithm, we use a modified version of the indicator function as proposed by [16]:

$$\mathbb{I}^{\mathrm{INC}}\{x \prec x'\} = \mathbb{I}\{x \prec x'\} + \frac{1}{2}\mathbb{I}\{x \perp x'\} \tag{4}$$

A more serious problem caused by incomparability is a complication of the variance estimation for $\widehat{S}(\mathbf{X}, \mathbf{Y})$ [16]. Therefore, it is not clear how Bernstein-like bounds [4], where the empirical variance estimate is used in the concentration inequality, could be applied.

[1] Although $\widehat{S}$ is a sum of n^2 random values, these values are combinations of only $2n$ independent values. This is why the convergence rate is not better than the usual one for a sum of n independent variables.

4.3 Preference-based racing algorithm

Our preference-based racing setup assumes K random variables $X_1, \ldots, X_K$ with distributions $\mathbf{P}_{X_1}, \ldots, \mathbf{P}_{X_K}$, respectively, and these random variables take values in a partially ordered set $(\Omega, \prec)$. Obviously, the value-based racing setup described in Section 2.3 is a special case, with $\Omega = \mathbb{R}$ and $\prec$ reduced to the standard $<$ relation on the reals (comparing rollouts in terms of their rewards). The goal of our preference-based racing (PBR) algorithm is to find the best κ random variables with respect to the surrogate decision model $\ll_C$ introduced in Section 3. This leads to the following optimization task:

$$\sum_{i \in I} \sum_{j \neq i} \mathbb{I}\{X_j \ll X_i\} \longrightarrow \max_{I \subseteq [K]:\ |I| = \kappa} \tag{5}$$

Thanks to the indicator function (4), we have $S(X_i, X_j) = 1 - S(X_j, X_i)$ and hence $\mathbb{I}\{X_i \ll X_j\} = \mathbb{I}\{S(X_i, X_j) > 1/2\} = \mathbb{I}\{S(X_j, X_i) < 1/2\}$, which simplifies our implementation.

Algorithm 2 shows the pseudocode of PBR. It assumes as inputs the number κ, an upper bound $n_{\max}$ on the number of realizations an option is allowed to sample, and an upper bound δ on the probability of making a mistake (i.e., returning a suboptimal selection). We will concisely write $s_{i,j} = S(X_i, X_j)$ and $\widehat{s}_{i,j}$ for its estimate. The confidence interval (3) of $\widehat{s}_{i,j}$ for confidence level $1 - \delta$ is denoted by $[\ell_{i,j}, u_{i,j}]$. The set A consists of those index pairs for which the preference can not yet be determined with high probability (i.e., $1/2 \in [\ell_{i,j}, u_{i,j}]$), but that are possibly relevant for the final outcome. Initially, A contains all K^2 pairs of indices (line 1).

PBR first samples each pair of options whose indices appear (at least once) in A (lines 4–5). Then, in lines 6–10, it calculates $\widehat{s}_{i,j}$ for each pair of options according to (2), and the confidence intervals $[\ell_{i,j}, u_{i,j}]$ based on (3).

Next, for each X_i, we compute the number z_i of random variables that are worse with high enough probability—that is, for which $u_{i,j} < 1/2, j \neq i$ (line 12). Similarly, for each option X_i, we also compute the number o_i of options X_j that are preferred to it with high enough probability—that is, for which $\ell_{i,j} > 1/2$ (line 13). Note that, for each X_j, there are always at most $K - z_j$ options that can be better. Therefore, if $\#\{j : K - z_j < o_i\} > K - \kappa$, then X_i is a member of the solution set I of (5) with high probability (see line 14). The indices of these options are collected in C. One can also discard options based on a similar argument (line 15); their indices are collected in D. Note that a selection or exclusion of an option requires at most K different confidence bounds to be bigger or smaller than $1/2$, and since we can select or discard an option at any time, the confidence level δ has to be divided by $K^2 n_{\max}$ (line 9).

In order to update A, we note that, for those options in $C \cup D$, it is already decided with high probability whether or not they belong to I. Therefore, if two options X_i and X_j both belong to $C \cup D$, then $s_{i,j}$ does not need to be sampled any more, and thus the index pair (i, j) can be excluded from A. Additionally, if $1/2 \notin [\ell_{i,j}, u_{i,j}]$, then the pairwise relation of X_i and X_j is known with high

Algorithm 2 PBR$(X_1, \ldots, X_K, \kappa, n_{\max}, \delta)$

1: $A = \{(i,j) |\ 1 \le i, j \le K\}$
2: $n = 0$
3: **while** $(n \le n_{\max}) \wedge (|A| > 0)$ **do**
4: **for all** i appearing in A **do**
5: $x_i^{(n)} \sim X_i$ ▷ Draw a random sample
6: **for all** $(i,j) \in A$ **do**
7: Update $\widehat{s}_{i,j}$ with the new samples according to (2)
8: using the indicator function $\mathbb{I}^{\mathrm{INC}}\{.,.\}$ from (4)
9: $c_{i,j} = \sqrt{\frac{1}{2n} \log \frac{2K^2 n_{\max}}{\delta}}$
10: $u_{i,j} = \widehat{s}_{i,j} + c_{i,j}\ ,\ \ell_{i,j} = \widehat{s}_{i,j} - c_{i,j}$
11: **for** $i = 1 \to K$ **do**
12: $z_i = |\{j :\ u_{i,j} < 1/2, j \neq i\}|$ ▷ Number of options that are beaten by i
13: $o_i = |\{j :\ \ell_{i,j} > 1/2, j \neq i\}|$ ▷ Number of options that beat i
14: $C = \left\{i : K - \kappa < \left|\{j : K - z_j < o_i\}\right|\right\}$ ▷ select
15: $D = \left\{i : \kappa < \left|\{j : K - o_j < z_i\}\right|\right\}$ ▷ discard
16: **for** $(i,j) \in A$ **do**
17: **if** $(i,j \in C \cup D) \vee (1/2 \notin [\ell_{i,j}, u_{i,j}])$ **then**
18: $A = A \setminus (i,j)$
19: ▷ Do not update $\hat{s}_{i,j}$ any more
20: $n = n + 1$
21: σ is a permutation that sorts the options in decreasing order based on $\widehat{d}_i = \#\{j \,|\, \ell_{j,i} > 1/2\}$.
22: **return** σ

enough probability, so (i,j) can again be excluded from A. These filter steps are implemented in line 17.

We remark that the terminal condition in line 3 is as general as possible and cannot be relaxed. Indeed, termination must be based on those preferences that are already decided (with high probability). Thus, assuming the options to be ordered according to $\ll_C$, the algorithm can only stop if $\min\{z_1, \ldots, z_\kappa\} \ge \max\{K - o_{\kappa+1}, \ldots, K - o_K\}$ or $\min\{o_{\kappa+1}, \ldots, o_K\} \le \max\{K - z_1, \ldots, K - z_\kappa\}$. Both conditions imply that $C \cup D = [K]$ and hence that A is empty.

4.4 Analysis of the PBR algorithm

Recall that PBR returns a permutation σ, from which the set of options B deemed best by the racing algorithm (in terms of $\ll_C$) can be obtained as $B = \{X_{\sigma(i)} | 1 \le i \le \kappa\}$. In the following, we consider the top-κ set B as the output of PBR.

In the first part of our analysis, we upper bound the expected number of samples taken by PBR. Our analysis is similar to the sample complexity analysis

of PAC-bandit algorithms [9].

Theorem 5. *For random variables $X_1, \ldots, X_K$,*

$$n_i = \left\lceil \frac{1}{2 \min_{j \neq i} \Delta_{i,j}^2} \log \frac{2K^2 n_{\max}}{\delta} \right\rceil \ ,$$

where $\Delta_{i,j} = S(X_i, X_j) - 1/2$. Then, whenever $n_i \leq n_{\max}$ for all $i \in [K]$, PBR *outputs the κ best options (with respect to $\ll_C$) with probability at least $1 - \delta$ and generates at most $\sum_{i=1}^{K} n_i$ samples.*

Proof. According to (3), for any i, j and round n, the probability that $s_{i,j}$ is not included in $[\ell_{i,j}, u_{i,j}]$ is at most $\delta/(K^2 n_{\max})$. Thus, with probability at least $1 - \delta/2$, $s_{i,j} \in [\ell_{i,j}, u_{i,j}]$ for every i and j throughout the whole run of the algorithm.

Similarly by (3), when for some i and j both X_i and X_j are sampled for at least n_i times, then $[\ell_{i,j}, u_{i,j}]$ contains $1/2$ with probability at most $\delta/(K^2 n_{\max})$. Furthermore, if for some i all the preferences against other options are decided (i.e., $\ell_{i,j} > 1/2$ or $u_{i,j} < 1/2$ for all j), then X_i will not be sampled any more.

Putting these observations together, the claim follows from the union bound. □

Remark 6. *We remark that Theorem 5 remains valid despite the fact that statistical independence is not assured, neither for the terms in $\widehat{s}_{i,j}$ nor for $\widehat{s}_{i,j}$ and $\widehat{s}_{i,j'}$ with $i, j, j' \in [K]$. First, the confidence interval of each $\widehat{s}_{i,j}$ is obtained based on the concentration property of the U-statistic (Theorem 4). Second, the confidence intervals of $\widehat{s}_{i,j}$ are calculated separately for all $i, j \in [K]$ in every iteration, and the subsequent application of the union bound does not require independence.*

In the second part of our analysis, we investigate the relation of the outcome of PBR to the decision model $\ll$. Theorem 5 and Proposition 3 have the following immediate consequence for PBR.

Theorem 7. *Let $\mathcal{X} = \{X_1, \ldots, X_K\}$ be a set of random variables with Smith set $\mathcal{C}^* \subseteq \mathcal{X}$. Then, under the conditions of Theorem 5, with probability at least $1 - \delta$,* PBR *outputs a set of options $B \subseteq \mathcal{X}$ satisfying the following: If $|\mathcal{C}^*| \leq \kappa$, then $\mathcal{C}^* \subseteq B$ (Smith efficiency), otherwise $B \subseteq \mathcal{C}^*$.*

Proof. The result follows immediately from Theorem 5 and Proposition 3. □

Thus, PBR finds the Smith set with high probability provided κ is set large enough; otherwise, it returns at least a subset of the Smith set. This indeed justifies the use of $\ll_C$ as a decision model. Nevertheless, as pointed out in Section 9 below, other surrogates of the $\ll$ relation are conceivable, too.

5 Alternative confidence intervals

In the preference-based racing framework, in order to being sample efficient, it is crucial to have a tight confidence bound for $\widehat{s}_{i,j}$ given in 2. We calculate a Bernstein-type bound for Wilcoxon two-sample statistic in Subsection 5.1 that makes us of the variance estimate of $\widehat{s}_{i,j}$. Then we will consider two different high-probability confidence intervals in Subsection 5.2 and 5.3 that are developed for binomial distribution and can be directly applied in our racing framework. Namely, we calculated a U-statistic based on the two observation sets whose terms are not independent. But if we compare them pairwisely, than we obtain a bionomially distributed sample set. Formally, given two sample set

5.1 Empirical Bernstein bound for Wilcoxon two-sample statistic

5.2 Clopper-Pearson exact confidence interval

5.3 Weissman bound

The result of [29](Theorem 2.1 therein) regarding L_1 deviation of empirical distribution can be adapted easily for multinomial distributions. Let us assume that we observed m times a multinomial distribution with parameter $\mathbf{p} \in \mathbb{R}^\ell$. The observations can be summarized as a histogram $(b_1, \ldots, b_\ell) \in \mathbb{N}^\ell$ where b_i denoted the number of times we observed the ith category, thus $\sum_{i=1}^{\ell} b_i = m$. Clearly, $\widehat{\mathbf{p}} = (\widehat{p}_1, \ldots, \widehat{p}_\ell)$ where $\widehat{p}_i = b_i/m$ is an estimate for $\mathbf{p}$. Then, according to [29], we have

$$\mathbf{P}\left(\|\mathbf{p} - \widehat{\mathbf{p}}\|_1 > \epsilon\right) \leq \left(2^\ell - 2\right) \exp\left(-\frac{m}{2}\epsilon^2\right) \tag{6}$$

By setting $\ell = 2$, we obtain a confidence interval for binomial distributions as

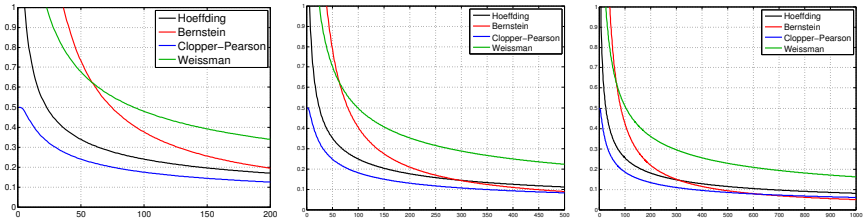
$$\|p - \widehat{p}\| \leq \sqrt{\frac{2}{n} \log \frac{2}{\delta}} \tag{7}$$

with probability at least $1 - \delta$. This bound is almost the same to the Hoeffding bound for two-sample statistics, but note that here p is the mean of m independent quantities.

5.4 Comparison of difference confidence intervals

6 Practical implementation of racing algorithms along with ES

In this section, we shall describe three practical considerations that makes the ES more efficient along with the racing framework.



(a) $n_{\max} = 200$ (b) $n_{\max} = 500$ (c) $n_{\max} = 1000$

Figure 2: The accuracy is plotted against the empirical sample complexities for the Hoeffding race algorithm (HR) and PBR, with the complexity parameter k shown below the markers. Each result is the average of 1000 repetitions. Information about the standard deviations for accuracy and sample complexity is provided in the right bottom corner.

First, the maximum size of roll-outs must be given in advance. But one can dynamically set the maximum size of roll-outs as it is proposed in [15].

To upper bound sample size by $K^2 n_{\max}$ is very conservative.

In the case of the ranking procedure we use, many estimates need to be decided that it is either significantly smaller than $1/2$ or significantly bigger. But if $s_{i,j}$ is close to $1/2$ or worse equal to $1/2$ then enormous number of sample is need to have a reliable enough estimate. Therefore we suggest a simple heuristic here, namely, we neglect those pairwise preferences that are close to $1/2$.

7 Experiments

In Section 7.1, we compare our PBR algorithm with the original Hoeffding race (HR) algorithm in terms of empirical sample complexity on synthetic data. In Section 7.2, we test our PB-EDPS method on a benchmark problem that was introduced in previous work on preference-based RL [7].

7.1 Results on synthetic data

Recall that our preference-based racing algorithm is more general than the original value-based one and, therefore, that PBR is more widely applicable than the Hoeffding race (HR) algorithm. This is an obvious advantage of PBR, and indeed, our preference-based generalization of the racing problem is mainly motivated by applications in which the value-based setup cannot be used. Seen from this perspective, PBR has an obvious justification, and there is in principle no need for a comparison to HR. Nevertheless, such a comparison is certainly interesting in the standard numerical setting where both algorithms can be used.

More specifically, the goal of our experiments was to compare the two algorithms in terms of their empirical sample complexity. This comparison, however,

has to be done with caution, keeping in mind that PBR and HR are solving different optimization tasks (namely (1) and (5), respectively): HR selects the κ best options based on the means, whereas the goal of PBR is to select κ options based on $\ll_C$. While these two objectives coincide in some cases, they may differ in others. Therefore, we considered the following two test scenarios:

1. *Normal distributions:* each random variable X_i follows a normal distribution $\mathcal{N}((k/2)m_i, c_i)$, where $m_i \sim U[0,1]$ and $c_i \sim U[0,1]$, $k \in \mathbb{N}^+$;

2. *Bernoulli distributions with random drift:* each X_i obeys a Bernoulli distribution $Bern(1/2) + d_i$, where $d_i \sim (k/10)\,U[0,1]$ and $k \in \mathbb{N}^+$.

In both scenarios, the goal is to rank the distributions by their means.[2] Note that the complexity of the racing problem is controlled by the parameter k, with a higher k indicating a less complex task; we varied k between 1 and 10. Besides, the following parameters were used in each run: $K = 10$, $\kappa = 5$, $n_{\max} = 300$, $\delta = 0.05$.

Strictly speaking, HR is not applicable in the first scenario, since the support of a normal distribution is not bounded; we used $R = 8$ as an upper bound, thus conceding to HR a small probability for a mistake.[3] For Bernoulli, the bounds of the supports can be readily determined.

Figure 3 shows the number of random samples drawn by the racing algorithms versus accuracy (percentage of true top-κ variables among the predicted top-κ). As we can see from the plots, PBR achieves a significantly lower sample complexity than HR, whereas its accuracy is on a par or better in most cases. While this may appear surprising at first sight, it can be explained by the asymptotic behavior of the statistics used. While HR uses the mean estimate whose variance decreases with $O(1/\sqrt{n})$, the variance for the Wilcoxon 2-sample statistic decreases as fast as $O(1/n^2)$, at least for $s_{i,j}$ close to to the extreme probabilities 0 or 1 [27].

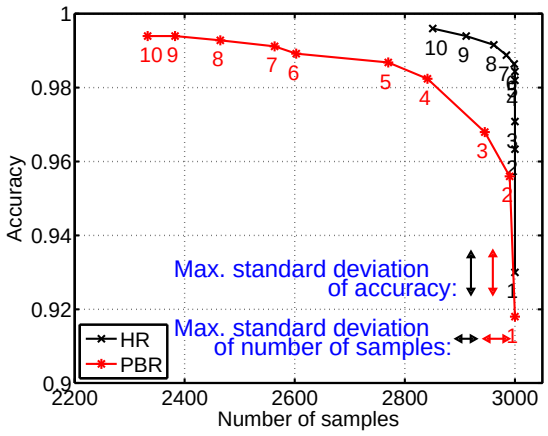
In the Bernoulli case, one may wonder why the sample complexity of PBR hardly changes with k (see the red point cloud in Figure 3(b)). This can be explained by the fact that the two sample U-statistic $\widehat{S}$ in (2) does not depend on the magnitude of the drift d_i (as long as it is smaller than 1).

7.2 Medical treatment design

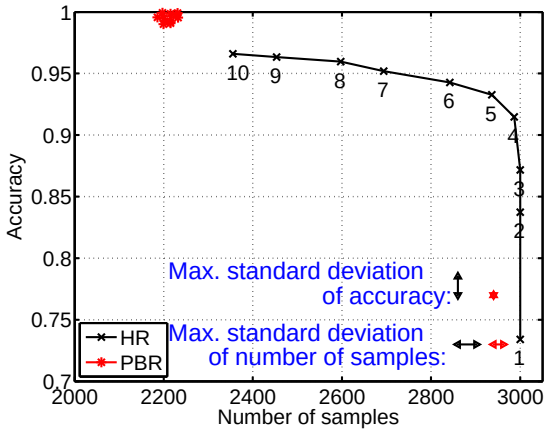
Here, we tackle a problem that has been used in previous work on preference-based RL [7, 3], namely the medical treatment design for cancer clinical trials.

[2] In order to show that the ranking based on means and $\ll$ coincide for a set of options $X_1, \ldots, X_K$ with means $\mu_1, \ldots, \mu_K$, it is enough to see that for any X_i and X_j, $\mu_i < \mu_j$ implies $S(X_i, X_j) > 1/2$. In the case of the normal distribution, this follows from the symmetricity of the density function. Now, let us consider two Bernoulli distributions with parameters p_1 and p_2, where $p_1 < p_2$. Then, a simple calculation shows that the value of $S(.,.)$ is $(p_2 - p_1 + 1)/2$, which is greater than $1/2$. This also holds if we add a drift $d_1, d_2 \in [0,1]$ to the value of the random variables.

[3] The probability that all samples remain inside the range is larger than 0.99 for $K = 10$ and $n_{\max} = 300$.



(a) Normal distributions



(b) Bernoulli distributions

Figure 3: The accuracy is plotted against the empirical sample complexities for the Hoeffding race algorithm (HR) and PBR, with the complexity parameter k shown below the markers. Each result is the average of 1000 repetitions. Information about the standard deviations for accuracy and sample complexity is provided in the right bottom corner.

The problem is to learn an optimal treatment policy π mapping states $\boldsymbol{s} = (S, X) \in \mathcal{S} = \mathbb{R}_+^2$ to actions in the form of a dosage level $d \in [0, 1]$; the drug is given once a month, and a patient is simulated over a fixed time horizon (we conducted two experiments with six and twelve months, respectively). A state $\boldsymbol{s} = (S, X)$ describes the health condition of the patient: S is the tumor size and X the level of toxicity, which is inversely related to the wellness of the patient. These two properties constitute conflicting criteria: An increase of the

dosage level will reduce the tumor size but increase toxicity and therefore affect the patient's wellness. A corresponding simulation model based on first-order difference equations was originally introduced in [32].

As argued by [7], the numerical rewards assigned to different health states of a patient (including the extreme case of death) are quite arbitrary in this model. Therefore, the authors propose an alternative and more realistic formalization, in which histories are compared in a qualitative way:

- Giving full priority to the survival of a patient, $\mathbf{h}' \preceq \mathbf{h}$ if the patient survives in $\mathbf{h}$ but not in $\mathbf{h}'$, and both histories are incomparable ($\mathbf{h}' \perp \mathbf{h}$) if the patient does neither survive in $\mathbf{h}'$ nor in $\mathbf{h}$.

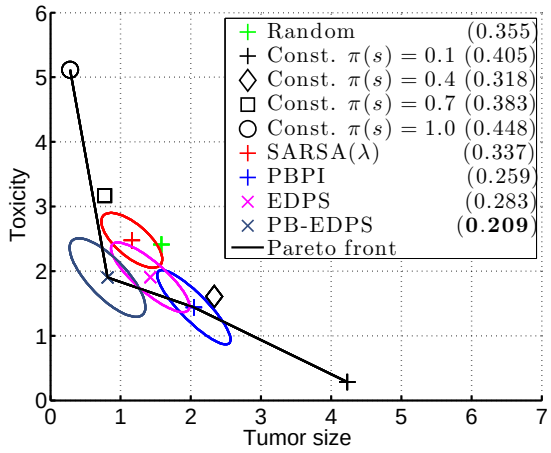
- Otherwise, if the patient survives in both histories, preference depends on the worst wellness of the patient and the final tumor size: Let C_X and C'_X denote, respectively, the *maximal* toxicity during the whole treatment in $\mathbf{h}$ and $\mathbf{h}'$, and C_S and C'_S the respective size of the tumor *at the end of the therapy*. Then, preference is defined via Pareto dominance: $\mathbf{h}' \preceq \mathbf{h}$ if (and only if) $C_X \leq C'_X$ and $C_S \leq C'_S$.

Let us again emphasize that $\preceq$ thus defined, as well as the induced strict order $\prec$, are only *partial* order relations. We used the same experimental setup as in previous work [7, 3], except for adding Gaussian noise $\mathcal{N}(0, 0.01)$ to the state observation [14], thereby making the underlying MDP partially observable.

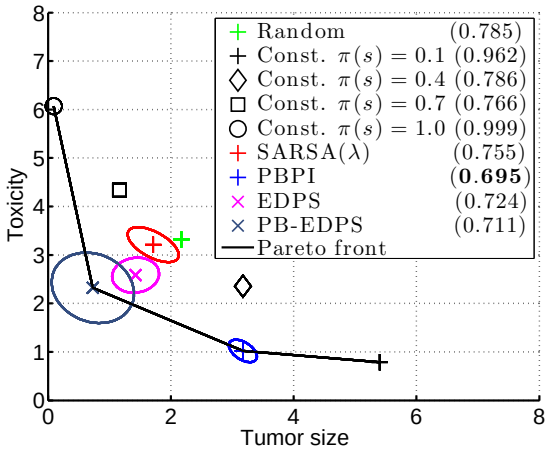
We run the implementation of [15] with the Hoeffding race algorithm and CMA-ES [13]; we refer to this implementation as EDPS. We set $\lambda = 6$ and $\mu = 3$ according to [13]. The initial global step size in CMA-ES was selected from $\{0.1, 1, 5, 10, 15, 25, 50, 100\}$. The racing algorithm has two hyperparameters, the confidence term δ and the maximum number of samples allowed for a single option, $n_{\max}$. We optimized δ in the range $\{0.01, 0.05, 0.1\}$, while $n_{\max}$ was initialized with 40 and then adapted using the technique of [15]. All parameter values were determined by means grid search, repeating the training process in each grid point (parameter setting) 100 times, and evaluating each model on 300 patients in terms of expected utility; we found $\sigma_0 = 2$, $\delta = 0.1$ to be optimal.

Our preference-based variant PB-EDPS as introduced in Section 3 was run with the same parameters. We used a sigmoidal policy space defined as $\pi_\Theta(\mathbf{s}) = 1/(1 + \exp(-\Theta^T \mathbf{s}))$. As baseline methods, we run the discrete uniform random policy (randomly choosing a dosage $d \in D' = \{0.1, 0.4, 0.7, 1.0\}$ each month) and the constant policies that take the same dosage $d \in D'$ independently of the patient's health state. As a more sophisticated baseline, we furthermore used SARSA(λ) [26] with discrete action set according to the original setup.[4] Finally, we included the preference-based policy iteration (PBPI) method of [12] with the parameters reported by the authors. Each policy learning method was run until reaching a limit 5000 training episodes.

[4] We used an ϵ-greedy policy for exploration. Initially, the learning rate α, the exploration term ϵ and the parameter of the replacing traces λ were set to 0.1, 0.2 and 0.95 respectively, and decreased gradually with a decay factor $1/\lceil \frac{10}{\tau} \rceil$, where τ is the number of training episodes. We discretized each dimension of the state space into 20 bins and used a tile coding to represent the action-value function. We refer to [28] for more details.



(a) 6 months treatment



(b) 12 months treatment

Figure 4: Illustration of patient status under different treatment policies. On the x-axis is the tumor size after 6 (a) and 12 (b) months, on the y-axis the highest toxicity during the treatment. The death rates are shown in parentheses at the upper right corner.

We evaluated each policy on 300 virtual patients and derived averages for C_X, the maximum toxicity level, as well as C_S, the tumor size at the end of the treatment. We repeated this process 100 times for each policy search method. Then, we plotted its mean and the 95% confidence regions (assuming a multivariate normal distribution), which represent the uncertainty coming from the repetitions of the training process. As can be seen in Figure 4, our approach is performing quite well and lies on the Pareto front of all methods

(which remains true when adding the death rate, reported in the same figure, as a third criterion).

8 Related Work

The idea of preference-based reinforcement learning was introduced simultaneously and independently in [2] and [7]. While preferences on trajectories (histories) are taken as a point of departure in both approaches, policy learning is accomplished in different ways. As already mentioned, Cheng et al. [7] generalize classification-based approximate policy iteration as proposed in [19, 20]. To this end, they train a model that ranks actions given a state, using a preference learning method called *label ranking*. Instead of ranking actions given states, Akrour at al. [2] exploit preferences on trajectories in order to learn a model that ranks complete policies. To this end, policies are mapped to real (feature) vectors that represent important properties of the induced trajectories. Then, a standard learning-to-rank method is applied in this behavioral representation space, in which preferences are expressed by an expert. Originally, this approach was motivated by concrete applications in robotics, whence the behavioral representation was specific to that application. In a follow-up work [3], the more generic behavioral representation of [1] was used, whereby a policy π is mapped to the frequency vector of states obtained by following that policy.

Evolutionary Direct Policy Search (EDPS) was introduced by Heidrich-Meisner and Igel [15], who used racing algorithms to control the number of rollouts in each iteration. Our approach can be viewed as a generalization of EDPS. Thanks to the preference-based racing algorithm we developed, it does not require access to the policy performances themselves but only to a ranking over them.

The racing setup and the Hoeffding race algorithm were first considered by [21, 22] in the context of model selection. For a detailed and precise implementation of the HR algorithm, see [15]. This algorithm was improved in [23], where the empirical Bernstein bound was used instead of the Hoeffding bound. In this way, the variance information of the mean estimates could be incorporated in the calculation of confidence intervals.

In the context of multi-armed bandits, a slightly different setup was introduced in [9], where an ϵ-optimal random variable has to be chosen with probability at least $1 - \delta$; here, ϵ-optimality of X_i means that $\mu_i + \epsilon \geq \max_{j \in [K]} \mu_j$. An algorithm solving this problem are called (ϵ, δ)-PAC bandit algorithm. The authors propose such an algorithm and prove an upper bound on the expected sample complexity. In this paper, we borrowed their technique and used it in the complexity analysis of PBR.

Recently, a PAC-bandit algorithm which is based on the widely-known UCB index-based muli-armed bandit method of [5] was introduced in [18]. In their formalization, an algorithm is an (ϵ, m, δ)-PAC bandit algorithm that selects the m best random variables under the PAC-bandit conditions. According to their definition, a racing algorithm is a $(0, \kappa, \delta)$-PAC algorithm. Instead of a high

probability bound for the *expected* sample complexity, the authors managed to prove such a bound for the *worst case* sample complexity. It is an interesting question whether or not the technique used in their proof, which makes use of a specific type of slack variables, can also be applied in our setting.

Yue et al. [31] introduced a multi-armed bandit setup where feedback is provided in the form of pairwise comparisons between options, just like in our approach. However, their decision model is more restrictive than ours. It not only assumes $\ll$ to be a total order, but also requires additional properties such as strong stochastic transitivity and stochastic triangle inequality.

9 Conclusion and Future Work

By introducing a preference-based extension of evolutionary direct policy search, called PB-EDPS, this paper contributes to the emerging field of preference-based reinforcement learning. Our method, which merely requires qualitative comparisons between sample histories as training information (and even allows for incomparability), is based on a theoretically sound decision-theoretic framework and shows promising results in first experimental studies.

The core of our method is a preference-based version of the Hoeffding race algorithm, for which we could provide theoretical guarantees. Empirically, we have seen that this algorithm is not only more widely applicable than the original value-based version, but may even reduce sample complexity in (numerical) settings where both versions can be used. Therefore, the idea of preference-based racing should not be limited to reinforcement learning; instead, it seems worthwhile to explore it for other applications, too, such as multi-objective optimization with several competing objectives [8].

Coming back to our PB-EDPS framework, we hope to achieve further improvements by elaborating on its individual components. For example, we would like to investigate the use of Bernstein instead of Hoeffding races, since Bernstein-like bounds (which exploit the empirical variance of the estimates) are normally tighter than Hoeffding bounds. Likewise, the Copeland relation $\ll_C$ is not necessarily an optimal surrogate of the $\ll$ relation on policies, and indeed, voting and decision theory offers a large repertoire of alternative relations that could in principle be used.

Our theoretical analysis so far essentially focused on the racing algorithm, and therefore only covers a single iteration of the evolutionary search process implemented by EDPS. Extending this analysis toward the convergence behavior of the complete search process is another important (and likewise difficult) topic to be addressed in future work.

Last but not least, there is also a need for further experimental studies, including both synthetic problems but also challenging real-world applications.

A Appendix

A.1 U statistics

The *U-statistics* play central role in many practical statistical problem. For an independent sample set $x^{(1)}, \ldots, x^{(n)}$ drawn from the same distribution over Ω, its general form can be written as

$$U = \frac{1}{\binom{n}{m}} \sum h\left(x^{(i_1)}, \ldots, x^{(i_m)}\right) \tag{8}$$

where the summation is taken over all subsets $\{i_1, \ldots, i_m\} \subseteq \{1, \ldots, n\}$ of size m, and where the kernel funcion h is of the form $\Omega^m \to [a, b]$ for some $a, b \in \mathbb{R}$. An especially attractive feature of this statistic is that it is an *efficient* (or *minimum variance unbiased*) estimator [27]. The U statistic also generalizes to multiple samples in a natural way. For example, the general form of the *two-sample U statistic* for two independent i.i.d. samples $x^{(1)}, \ldots, x^{(n)}$ and $y^{(1)}, \ldots, y^{(n')}$ drawn from Ω and Ξ, is

$$U = \frac{1}{\binom{n}{m}\binom{n'}{m'}} \sum h\left(x^{(i_1)}, \ldots, x^{(i_m)}, y^{(i'_1)}, \ldots, y^{(i'_{m'})}\right) \tag{9}$$

where the summation is taken over all subsets $\{i_1, \ldots, i_m\} \subseteq \{1, \ldots, n\}$ of size m and $\{i'_1, \ldots, i'_{m'}\} \subseteq \{1, \ldots, n'\}$ of size m', and, similalry, the kernel function h is a bounded function of the form $\Omega^m \times \Xi^{m'} \to [a, b]$ for some $a, b \in \mathbb{R}$. The general form of the Hoeffding theorem for two-sample U statistics can be written as follows.

Theorem 8 ([17], §5b)**.** *For any $\epsilon > 0$, using the notations introduced above,*

$$\mathbf{P}\left(|U - \mathbb{E}[U]| \geq \epsilon\right) \leq 2 \exp\left(\frac{-2k\epsilon^2}{(b-a)^2}\right)$$

where U is defined as in (9) and

$$k = \min\left(\lfloor n/m \rfloor \lfloor n'/m' \rfloor\right)$$

We applied Theorem 8 to Wilcoxon two-sample statistic in Subsection 4.1.

References

[1] Abbeel, P., Ng, A.: Apprenticeship learning via inverse reinforcement learning. In: Proceedings of the 21th International conference on Machine Learning, pp. ??–?? (2004)

[2] Akrour, R., Schoenauer, M., Sebag, M.: Preference-based policy learning. In: Proceedings ECMLPKDD 2011, European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, pp. 12–27 (2011)

[3] Akrour, R., Schoenauer, M., Sebag, M.: April: Active preference-learning based reinforcement learning. In: Proceedings ECMLPKDD 2012, European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, pp. 116–131 (2012)

[4] Audibert, J., Munos, R., Szepesvári, C.: Tuning bandit algorithms in stochastic environments. In: Proceedings of the Algorithmic Learning Theory, pp. 150–165 (2007)

[5] Auer, P., Cesa-Bianchi, N., Fischer, P.: Finite-time analysis of the multiarmed bandit problem. Machine Learning **47**, 235–256 (2002)

[6] Beyer, H., Schwefel, H.: Evolution strategies–a comprehensive introduction. Natural computing **1**, 3–52 (2002)

[7] Cheng, W., Fürnkranz, J., Hüllermeier, E., Park, S.: Preference-based policy iteration: Leveraging preference learning for reinforcement learning. In: Proceedings ECMLPKDD 2011, European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, pp. 414–429 (2011)

[8] Coello, C., Lamont, G., Van Veldhuizen, D.: Evolutionary algorithms for solving multi-objective problems. Springer (2007)

[9] Even-Dar, E., Mannor, S., Mansour, Y.: PAC bounds for multi-armed bandit and markov decision processes. In: Proceedings of the 15th Annual Conference on Computational Learning Theory, pp. 255–270 (2002)

[10] Fishburn, P.: Nontransitive measurable utility. J. Math. Psychology **26**, 31–67 (1982)

[11] Fürnkranz, J., Hüllermeier, E. (eds.): Preference Learning. Springer-Verlag (2011)

[12] Fürnkranz, J., Hüllermeier, E., Cheng, W., Park, S.: Preference-based reinforcement learning: a formal framework and a policy iteration algorithm. Machine Learning **89**(1-2), 123–156 (2012)

[13] Hansen, N., Kern, S.: Evaluating the CMA evolution strategy on multimodal test functions. In: Parallel Problem Solving from Nature-PPSN VIII, pp. 282–291 (2004)

[14] Heidrich-Meisner, V., Igel, C.: Variable metric reinforcement learning methods applied to the noisy mountain car problem. Recent Advances in Reinforcement Learning pp. 136–150 (2008)

[15] Heidrich-Meisner, V., Igel, C.: Hoeffding and Bernstein races for selecting policies in evolutionary direct policy search. In: Proceedings of the 26th International Conference on Machine Learning, pp. 401–408 (2009)

[16] Hemelrijk, J.: Note on Wilcoxon's two-sample test when ties are present. The Annals of Mathematical Statistics **23**(1), 133–135 (1952)

[17] Hoeffding, W.: Probability inequalities for sums of bounded random variables. Journal of the American Statistical Association **58**, 13–30 (1963)

[18] Kalyanakrishnan, S., Tewari, A., Auer, P., Stone, P.: Pac subset selection in stochastic multi-armed bandits. In: Proceedings of the Twenty-ninth International Conference on Machine Learning (ICML 2012), pp. 655–662 (2012)

[19] Lagoudakis, M., Parr, R.: Reinforcement learning as classification: Leveraging modern classifiers. In: Proceedings of the 20th International Conference on Machine Learning, pp. 424–431 (2003)

[20] Lazaric, A., Ghavamzadeh, M., Munos, R.: Analysis of a classification-based policy iteration algorithm. In: Proceedings of the 27th International Conference on Machine Learning, pp. 607–614 (2010)

[21] Maron, O., Moore, A.: Hoeffding races: accelerating model selection search for classification and function approximation. In: Advances in Neural Information Processing Systems, pp. 59–66 (1994)

[22] Maron, O., Moore, A.: The racing algorithm: Model selection for lazy learners. Artificial Intelligence Review **5**(1), 193–225 (1997)

[23] Mnih, V., Szepesvári, C., Audibert, J.: Empirical Bernstein stopping. In: Proceedings of the 25th international conference on Machine learning, pp. 672–679 (2008)

[24] Moulin, H.: Axioms of cooperative decision making. Cambridge University Press (1988)

[25] Puterman, M.: Markov decision processes: discrete stochastic dynamic programming. John Wiley & Sons, Inc. (1994)

[26] Rummery, G.A., Niranjan, M.: On-line Q-learning using connectionist systems. Tech. Rep. CUED/F-INFENG/TR 166, Cambridge University, Engineering Department (1994)

[27] Serfling, R.: Approximation theorems of mathematical statistics, vol. 34. Wiley Online Library (1980)

[28] Szepesvári, C.: Algorithms for reinforcement learning. Morgan and Claypool (2010)

[29] Weissman, T., Ordentlich, E., Seroussi, G., Verdu, S., Weinberger, M.J.: Inequalities for the l1 deviation of the empirical distribution. Tech. Rep. HPL-2003-97 (R.1), HP Laboratories Palo Alto (2003)

[30] Williams, R.: Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine Learning **8**(3), 229–256 (1992)

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

[32] Zhao, Y., Kosorok, M., Zeng, D.: Reinforcement learning design for cancer clinical trials. Statistics in Medicine **28**(26), 3294–3315 (2009)