Nonparametric Contextual Bandit Optimization via Random Approximation

Craig Corcoran **Harsh Pareek** **Ian Yang** **Pradeep Ravikumar**

Abstract

We examine the stochastic contextual bandit problem in a novel continuous-action setting where the policy lies in a reproducing kernel Hilbert space (RKHS). This provides a framework to handle continuous policy and action spaces in a tractable manner while retaining polynomial regret bounds, in contrast with much prior work in the continuous setting. We extend an optimization perspective that has recently been popular in the bandit literature to the nonparametric bandit setting which naturally encompasses both finite and continuous-action contextual bandit problems. Optimization frameworks naturally balance between exploration and exploitation reducing the need for parameter tuning. Building on prior work in random function approximation for kernel methods, we show a new approximation error bound with linear convergence in the case where the loss $\ell(s, a)$ is smooth and strongly convex in the action a for all contexts s. Combining this result with prior work on finite-dimensional online bandit optimization, we present a fast algorithm for online nonparametric contextual bandit optimization with an $O\left(\sqrt{\log^3(T)/T}\right)$ average regret guarantee that holds against an oblivious adversary, the first regret bound in this setting.

1 Introduction

Reinforcement learning is concerned with designing algorithms (or learning agents) for sequential decision making tasks, where the agent learns from experience gained through interacting with the world in the form of *evaluative* (or bandit) feedback (**?**). In the evaluative feedback model, the learning agent makes a decision and is presented with an assessment of the decision it has made but is given *no* information on the quality of other alternatives it could have chosen. This kind of feedback should be contrasted with that in the supervised learning setting where the learning agent has access to information about the quality of *each* of the possible alternatives via supervised labels.

There is a spectrum of assumptions made in the literature about the information available to the agent in a reinforcement learning task. On one end of this spectrum are Multi-Armed Bandit (MAB) problems where in each round the learner is presented a finite number of actions (arms) which give (stochastic or adversarial) rewards when "pulled". On the other end of the spectrum are the Markov Decision Process (MDP) problems in which the agent observes a sequence of states dependent on the previous state(s) and the action taken by the agent, thus the agent has some control over which states are visited and must consider the long-term consequences of each action.

An interesting problem in between these two extremes is the Contextual Bandit which serves as the motivation for this paper. At each time step, the agent is presented with a state, also referred to as the context, which is chosen independently of all previous actions. The reward (or loss) for an action depends on the context; thus, a learner must learn a mapping between the contexts and the actions identifying the optimal action for each context. This problem is similar to online supervised learning where the goal is to learn a mapping from inputs to given outputs, with the key difference being that feedback is *evaluative*. The contextual bandit framework applies naturally to problems such as online recommendations and clinical trials.

At first glance, contextual bandits appear to be fundamentally different from simple bandit problems such as the MAB. However, they can be seen as a natural extension of MAB, where each potential policy, i.e. mapping from contexts to actions, is an arm and the agent, instead of pulling arms corresponding to actions, is pulling arms corresponding to *policies* and taking whatever action the policy suggests under the context of that round.

Applying the MAB framework directly leads to a number of technical problems: regret bounds for multi armed bandits depend explicitly on the number of arms involved, making them intractable in settings where there are infinitely many policies, such as the contextual bandit problem with a continuous state or action space. Additionally, the MAB framework assumes that the rewards of the arms are all independent of each other, which is not the case in most problems with an infinite number of policies where small variations in the policy typically produce small changes in the reward.

Such considerations have led to the development of continuum-armed bandits eg.(**?**), where some structure relating the arms to one another is assumed. However, current approaches to continuum-armed contextual bandits are unsatisfactory for our setting. They often make very strong assumptions such as a finite-dimensional policy or reward function parameterization, requiring a large degree of domain knowledge, or they make very weak assumptions, such as only using Lipschitz continuity, resulting in guarantees that suffer from the curse of dimensionality.

This motivates us to study convex nonparametric bandit problems as defined in this paper, where the arms (or policies) belong to a Reproducing Kernel Hilbert Space (RKHS). This also naturally allows us to study continuous action contextual bandit problems where the arms themselves form a metric space, a setting which has received less attention in the literature.

While few theoretical results exist for continuous action MDPs, the approaches used in practice such as policy search algorithms are closer in spirit to our algorithms than they are to procedures previously analyzed in the contextual bandit literature. This provides additional motivation for our setting and a direction for future work.

The contributions of this paper are as follows: we present a novel formulation of continuous action contextual bandits problem as a nonparametric zeroth-order optimization problem, providing the first regret bounds in this setting. Furthermore, our bounds are independent of the state or action dimension and the algorithm is computationally efficient, making the approach attractive in the high-dimensional setting. Along the way, we extend the coordinate descent analysis in **?** for random features to the case where the objective is smooth and and strongly-convex with respect to the actions, providing new "linear" $O(c^d)$ approximation error bounds. This result is independent of the contextual bandit problem and may be of general interest. Finally, we demonstrate our algorithm on contextual bandit benchmark tasks where it outperforms the parametric and finite-action baselines.

2 Related Work

Our analysis builds upon two areas of prior work:

The first is the optimization approach to stochastic bandit learning problems, which utilizes the equivalence between evaluative (or bandit) feedback and zeroth order oracle access as studied in the optimization literature (**??**). This approach estimates the gradient from (single-point) function evaluations alone, allowing the use of efficient gradient descent algorithms. In Section **??**, we review several fundamental bandit problems and discuss them from an optimization perspective, surveying their oracle complexity, as summarized in Table **??**. In Section **??**, we discuss prior work on zeroth order optimization in greater detail.

The second is an approach for scaling nonparametric kernel learning to very large problems using random features sampled from a distribution defined by the kernel. In Section **??** we summarize the key results from this literature which we will use later.

Notation We use K to denote the number of arms or actions for a bandit (if finite), A or $\mathcal{A}$ to denote the set of discrete arms or continuous actions respectively, and T to denote the number of rounds or function evaluations. The dimension of the relevant continuous optimization problem is denoted as d. Δ^d is the d-dimensional simplex. $\mathcal{H}_k$ denotes the RKHS defined by the kernel $k(\boldsymbol{x}, \boldsymbol{y})$.

Problem	Actions	Loss	Domain	Average Regret
Bandit	Finite	Linear	Δ^K	$O(\sqrt{K/T})$
Bandit	Continuous	Smooth, Str. Convex	$\mathbb{R}^d$	$O(\sqrt{d^2/T})$
Contextual	Finite	Linear	$\Delta^{\lvert\Pi\rvert}$	$O(\sqrt{K \log \lvert\Pi\rvert/T})$
Contextual	Continuous	Smooth, Str. Convex*	$\mathcal{H}_k$	$O(\sqrt{\log^3(T)/T})$

Table 1: *Loss is $\sum_{t=1}^{T} \ell(\boldsymbol{s}_t, \boldsymbol{a})$ where $\ell(\boldsymbol{s}_t, \boldsymbol{a})$ is smooth and strongly convex in $\boldsymbol{a}$ for all t. This table compares the structure and domain of the loss function for multiple bandit problems popular in the literature against our setting and the corresponding optimization bounds. The finite dimensional results are compiled from **???**. The bound for the nonparametric continuous-action contextual bandit problem given in the last row is the primary contribution of this work.

2.1 Review of Bandit Problems, an Optimization View

In general, bandit problems are online learning problems where information is obtained through evaluative feedback. The classical example of the stochastic Multi-Armed Bandit described below emphasizes this nature of the problem. When one arm of the bandit is pulled, it gives a noisy reward which reveals information about its reward distribution but not that about other arms. Many approaches to bandit problems involve distinct exploration and exploitation stages; the goal of the exploration stage being eliciting the expected reward for all arms while that of the exploitation stage being pulling the best known arm (so far) to maximize the reward. These algorithms thus exhibit an explicit exploration-exploitation tradeoff and include a hyperparameter to control this tradeoff. As we discuss below, optimization approaches naturally incorporate both goals simultaneously.

Multi-Armed Bandit The stochastic multi-armed bandit (MAB) problem consists of a set of discrete actions (or arms) A, where actions have fixed, independent loss (or reward) distributions $p_i(\ell)$. In the literature both reward and loss formulations are used, but in this paper we use the loss or regret formulation. The goal of the algorithm is to "pull" arms in a sequence that minimizes the total loss: $\min \sum_{t=1}^{T} \mathbb{E}[\ell(a_t)]$.

The MAB problem can be posed as a linear optimization problem over the simplex Δ^K by considering the actions $x \in \Delta^K$ to be distributions over A rather than the discrete actions $a \in A$ themselves. Then, $\mathbb{E}[\ell(x)] = c^T x$, where $c_i = \mathbb{E}[\ell(a_i)]$, and each round the agent is given access to noisy observations of c_i when taking action a_i. Thus, bandit problems perform linear optimization in the zeroth order setting where the linear function is not explicitly known and only available through noisy evaluation at the vertices. In each round, the agent samples an arm according to this distribution and the result of playing this mixture gives an unbiased noisy estimate of the value at that point inside the simplex. In this setting the average regret is known to be $\Theta(\sqrt{K/T})$ in general and $O(K \log(T)/T)$ if the gap between the best and second-best arm is bounded away from zero (**??**). Linear functions over more general convex domains have been studied in **?** and **?**, with bounds ranging from $O(\sqrt{d/T})$ to $O(\sqrt{d^2/T})$.

Continuum-armed Bandits The simplex formulation of MAB as above can be generalized to optimizing a nonlinear convex loss over an n-dimensional convex domain where the arms are points in this domain. In the case of a linear loss, the optimal points are known to lie on the boundary, however this does not hold for nonlinear convex losses. The nonlinear convex case was first discussed in **?**. More recently, average regret bounds of $O(\sqrt[4]{d^2/T})$, $O(\sqrt[3]{d^2/T})$, and $O(\sqrt{d^2/T})$ have been shown for convex, strongly convex, and strongly convex and smooth functions respectively (**???**). The algorithms used to achieve these bounds are variants of stochastic gradient descent where the gradient is approximated using function evaluations provided by a noisy zeroth order oracle.

The more general nonconvex setting in which the loss function is Lipschitz-continuous and the actions form a metric space was studied in **??**. **?** make the related assumption that the cost distribution is a Gaussian process, giving bounds in terms of effective dimension. However, the nonconvexity of these settings causes their bounds to grow exponentially with the dimension. This nonconvex, continuous loss setting is sometimes called "nonparametric" because these approaches make nonparametric

assumptions about the form of the loss function. We note a distinction as to what part of the problem is being treated nonparametrically. In our work the *object being optimized* is nonparametric (infinite dimensional), while in the nonparametric works listed above the *loss function* is nonparametric, but the object being optimized is parametric (finite dimensional).

Contextual Bandits The contextual bandit problem extends the MAB setting by including a context or state $s \in S$ at each round; the loss distribution $\ell(s, a) \sim p(\ell|s, a)$ is now a function of both the state s provided by the environment and the action a chosen by the algorithm. Thus the agent must learn a mapping from states to actions, which we will call the policy π.

The contextual bandit problem with a context (or state) s where the loss is a function of the state and action $\ell(s, a)$ and the action set is finite was first analyzed in **?**, where they examined the two-armed contextual bandit problem. In the finite K-action case, a common approach is to define the policy as a distribution over a finite set of "expert" policies Π. With this assumption, **??** give optimal $O(\sqrt{K \log |\Pi|/T})$ regret bounds, though the computational complexity of their EXP4 algorithm is linear in $|\Pi|$, making it intractable for very large policy classes. **?** address this computational issue using a coordinate descent procedure that maintains a sparse distribution over the experts. Their algorithm assumes access to an "argmax oracle" which can return the best expert in Π based on the current set of samples in constant time. Stated as an optimization problem, the experts approach to contextual bandits is very similar to the MAB case where we perform stochastic zeroth order optimization over the simplex of experts.

Alternatively, **?** shows an algorithm that can compete with a parametric set of policies of VC dimension d, obtaining $O(\sqrt{d \log(T)/T})$ regret bounds in the discrete action setting. Similar regret bounds are obtained for contextual bandits with the linear loss assumption (**?**). Also worth noting is **?** which uses kernel methods to represent the state-action loss is a nonparametric way, but it still relies on a finite action set. Again, we contrast this with our approach where the actions are continuous and the policy is nonparametric.

There has been less work on contextual bandits for continuous action spaces. **?** examines the case where the (state, action) loss is Lipschitz continuous. With only this assumption, the analysis again results in bounds that scale exponentially with the dimension in the worst case. While parameterizing the loss function is a common assumption, as in the linear bandits above, it is less common to parameterize the policy directly. **?** present an actor-critic architecture where both the policy and the loss function are parameterized independently. Parameterized policies are more common in the control and reinforcement learning communities (see seminal works **??** or the more recent survey **?**). They deal with the more difficult Markov Decision Process setting where the agent's actions affect the next state distribution, and any guarantees are for asymptotic convergence to a local optimum.

To summarize, prior work on contextual bandits has made more restrictive simplifying assumptions on the form of the policy, such as assuming a finite-dimensional (e.g. linear) parameterization for ℓ or π, a notable one being a mixture of experts setting as in (**?**). However, without extra knowledge that the optimal policy lies in some finite-dimensional parameterized class, the policy mapping is naturally an infinite-dimensional object. Thus the contextual bandit problem is most naturally posed as nonparametric bandit learning.

2.2 Zeroth Order Optimization

Optimal algorithms for general finite-dimensional convex bandit optimization typically perform some form of Stochastic Gradient Descent (SGD). SGD is a very popular algorithm because it is fast, easy to implement, and has been shown to have optimal $O(1/\sqrt{T})$ regret (**???**) in the first order oracle setting where the algorithm is given direct access to an unbiased estimate of the gradient. An interesting property of SGD is that the bounds are dimension *independent*, under a dimension-independent bound on the variance of the gradient estimator, an attractive property for high dimensional problems.

In the full-information (supervised) setting, the agent is given complete access to the loss $\ell(s, a)$ for all actions a and can thus easily compute the gradient w.r.t. a. **?** explore the spectrum of problems between the bandit and the supervised setting, by considering multi-point feedback from a particular loss ℓ_t on round t. However, in many problems of interest—such as when the loss distribution is the result of noisy observations or the typical online learning setting—the agent is only allowed single point feedback, thus this work does not apply. Instead we use the single-point gradient estimator of the form introduced in **?**:

$$\nabla f(x) \approx \mathbb{E}_u \left[\frac{d}{\delta} f(x + \delta u) u \right] \tag{1}$$

where δ is a scalar and u is sampled from the uniform distribution over the unit sphere. They show $O(1/T^{1/4})$ average regret bounds that hold even in the oblivious adversarial setting. This rate can be improved to $O(1/T^{1/3})$ for the smooth and strongly convex case, and further to $O(1/\sqrt{T})$ if additionally the domain is unconstrained (**?**). This single-point gradient estimator plays a central role in our nonparametric bandit algorithms. Unfortunately, the variance of these gradient estimators depends on the dimension d, which in our case is infinite, and thus we cannot perform zeroth order optimization directly. However, we show that we can exploit the RKHS assumption on the space of policies to avoid this issue.

Worth noting also are approaches which avoid the difficulty of estimating the gradient from function values by performing a form of search using a separation oracle, such as the ellipsoid algorithm (**??**), or the random walk approach in **?**. They achieve $\tilde{O}(poly(d)/\sqrt{T})$ optimization error or average regret respectively, which is optimal in terms of its scaling with T. However, these algorithms scale poorly with the dimension.

2.3 Random Features for Scalable Nonparametric Learning

Kernels are a general way to provide structure to a function class by defining a notion of smoothness without making more restrictive (parametric) assumptions. Recently, random features have been used to improve the computational performance of nonparametric kernel methods for large problems requiring many samples (**???**). This approach relies on the following theorem, which shows that any positive definite (PD) kernel can be decomposed in terms of a feature map and a measure over the set of features.

Theorem 1. *From* **??***: If $k(x, y)$ is a PD kernel, then there exists a set Θ and a measure p on Θ, and features parameterized by θ, $\phi(x; \theta) : \mathcal{X} \mapsto \mathbb{R}$ in $L_2(\mathcal{X})$, such that*

$$k(x, y) = \int p(\theta)\phi(x; \theta)\phi(y; \theta)d\theta \tag{2}$$

Thus sampling random features according to p will in expectation approximate functions in the corresponding RKHS well. Using a randomized coordinate descent analysis in the infinite dimensional Hilbert space, **?** show that sampling d features according to $p(\theta)$ gives $O(1/d)$ expected approximation error when the loss is smooth and convex. A major contribution of this work–which may be of general interest–is to extend this coordinate descent analysis to the case where the loss is a sum over functions $\ell(s_t, a)$ which are smooth and strongly convex in a for all t, resulting in exponentially decaying approximation error bound $O(a^d)$.

In this paper, the kernel is defined as TODO. In the general case when the state and action spaces are not one-dimensional and unconstrained, we can replace the analysis with TODO

3 Nonparametric Contextual Bandits Optimization

Recall that in the contextual bandit problem the agent wants to find a policy π that minimizes the (average) cumulative loss over all rounds:

$$\min_{\pi} \ \frac{1}{T} \sum_{t=1}^{T} \ell(s_t, \pi(s_t)) \tag{3}$$

Then, following the optimization perspective on bandit learning as discussed in Section **??**, the agent can be modelled as directly solving this optimization problem. In round t the agent plays a policy π_t which can be viewed as a query to a zeroth-order oracle for (**??**). This suggests zeroth order gradient descent as a viable strategy for this problem. However, note that for problems with continuous state or action spaces, the space of all policies is infinite-dimensional. As mentioned in Section **??**, the bounds for zeroth order gradient descent involve the dimension explicitly and would not give finite regret bounds!

We address this issue directly by formulating (**??**) as a nonparametric optimization problem in an RKHS, rather than making parametric assumptions on the form of the reward function or the policy. Using a random feature approximation approach, we show an $O\left(\sqrt{\frac{\log^3 T}{T}}\right)$ average regret bound when $\ell(s, a)$ is smooth and strongly convex in a for all $s \in S$ utilizing a coordinate descent analysis along the lines of **?** and Constant Nullspace Strong Convexity (CNSC) (**?**) arguments.

We assume that we are given access to a zeroth-order oracle which at each round t presents the algorithm with a state s_t and then reveals the loss $\ell(s_t, a_t)$ for the action a_t chosen by the algorithm. The oracle may choose the sequence of states $s_{1:T}$ adversarially, but is oblivious to the algorithm's internal randomization. We also assume we are provided with a kernel decomposition for the kernel $k(x, y)$ in terms a set of features $\phi(x; \theta)$ and a distribution over the feature parameters $p(\theta)$ as in equation (**??**). Given this setting, the algorithm competes against policies in the RKHS defined by $k(x, y)$ of the form $\pi(s) = \langle w(\theta), \phi(s; \theta)\rangle$. The problem is therefore an infinite-dimensional optimization over the space $\boldsymbol{w}$ of policies:

$$\min_{\boldsymbol{w} \in \mathcal{H}_k} \mathcal{L}(\boldsymbol{w}) \triangleq \sum_{t=1}^{T} \ell(s_t, \langle \boldsymbol{w}, \boldsymbol{\phi}(s_t)\rangle). \tag{4}$$

Given this problem setting, the general strategy of our algorithm is simple: first sample $d(T)$ features from $p(\theta)$ according to the given number of rounds T, then perform standard (finite-dimensional) zeroth order gradient descent in this d-dimensional space (see Algorithm **??**). As the algorithm is only given function values–not the gradient directly–we estimate the gradient using the single-point gradient estimator in equation (**??**), with the slight modification that we subtract the average function value from the previous gradient iteration from the current function value when estimating the gradient. This modification reduces variance and still provides an unbiased estimator as the expectation of u is zero. The single-point estimator is essential as the algorithm is only allowed to see the result of one action in each state before being presented with the next state and so does not have multi-point access to $\mathcal{L}_t$, as is necessary for the algorithms presented in **?** and **?**.

Algorithm 1 Random Feature Bandit Gradient Descent

Given: number of features d, random feature class $\phi(x, \theta)$, feature distribution $p(\theta)$, learning rate schedule $\{\eta_t\}_{t=1}^T$, sampling radius schedule $\{\delta_t\}_{t=1}^T$.
Notation: $\mathbb{S}$ is the unit sphere, $U_{\mathbb{S}}$ is the uniform distribution over $\mathbb{S}$.
$\theta_i \sim p(\theta)$, $i = 1, 2, \ldots, d$
$\boldsymbol{\phi}_d(x) = [\phi(x; \theta_1), \ldots, \phi(x; \theta_d)]^T$
$\boldsymbol{w}_0 = \mathbf{0}$
for $t = 1, 2, ..., T$ **do**
  $\boldsymbol{u} \sim U_{\mathbb{S}}$
  $f_t \leftarrow \ell_t(s_t, \langle \boldsymbol{w}_{t-1} + \delta \boldsymbol{u}, \boldsymbol{\phi}_d\rangle)$
  $\boldsymbol{g}_t \leftarrow (f_t - f_{t-1})\boldsymbol{u}$
  $\boldsymbol{w}_t \leftarrow \boldsymbol{w}_{t-1} - \eta_t \boldsymbol{g}_t$
end for
return $\boldsymbol{w}_T$

To the best of our knowledge, zeroth order convex optimization in an infinite dimensional setting has not previously been addressed in the literature. In the finite-dimensional setting, the regret bounds depend explicitly on the problem dimension, and as a result, are not readily applied to the infinite-dimensional case. However, as we show in this paper, the case where the domain being optimized over is an RKHS, is indeed tractable. In particular, we show in section **??** that the regret for policy in RKHS only increases by a poly-logarithmic factor in T compared to the parametric case.

3.1 Convergence Results

Notation First, we will lay out some notation used in this section. The average loss over all T round is expressed as $\mathcal{L}(\boldsymbol{w}) \triangleq \frac{1}{T}\sum_{t=1}^{T} \mathcal{L}_t(\boldsymbol{w})$ where $\mathcal{L}_t(\boldsymbol{w}) \triangleq \ell(s_t, \langle \boldsymbol{w}, \boldsymbol{\phi}(s_t)\rangle)$. We denote the average regret as $R(T) \triangleq \frac{1}{T}\sum_{t=1}^{T} \mathcal{L}_t(\boldsymbol{w}_t) - \mathcal{L}(\boldsymbol{w}^*)$. The feature weight function giving an optimal

policy in the RKHS is written as $\boldsymbol{w}^* \in \arg\min_{\boldsymbol{w}\in\mathcal{H}_k} \mathcal{L}(\boldsymbol{w})$ where $\boldsymbol{w} \in \mathcal{H}_k$ is understood to mean that $\langle \boldsymbol{w}, \boldsymbol{\phi}\rangle \in \mathcal{H}_k$. We use the *average* regret formulation as opposed to the total regret often used in the literature to emphasize the connection to optimization results.

We wish to bound the average regret $R(T)$ relative to the best policy in $\mathcal{H}_k$. Our analysis splits the regret into two components: the approximation error from using a finite set of features, and the optimization error from the approximate optimization via bandit gradient descent w.r.t. the random features.

After sampling the set of random features $\Theta = \{\theta_j\}_{j=1}^d$, the algorithm searches the subspace of functions spanned by the d features, denoted $\Pi_d = \{\pi(s) : \pi(s) = \langle \boldsymbol{w}, \boldsymbol{\phi}(s)\rangle, supp(\boldsymbol{w}) \in \Theta\}$. The approximation error is the difference in loss between π^* and

$$\pi_d^* \in \arg\min_{\pi\in\Pi_d} \sum_{t=1}^{T} \ell(s_t, \pi(s_t)) \tag{5}$$

where $\pi_d^*(s) = \langle \boldsymbol{w}_d^*, \boldsymbol{\phi}(s)\rangle$. We can then express the optimization error as the sum of two error components:

$$\mathcal{L}(\boldsymbol{w}_t) - \mathcal{L}(\boldsymbol{w}^*) = \underbrace{\mathcal{L}(\boldsymbol{w}_d^*) - \mathcal{L}(\boldsymbol{w}^*)}_{\text{approximation error}} + \underbrace{\mathcal{L}(\boldsymbol{w}_t) - \mathcal{L}(\boldsymbol{w}_d^*)}_{\text{optimization error}}$$

Linear Convergence of Random Features First, we derive an exponentially decaying bound on the estimation error $\mathcal{L}(\boldsymbol{w}_d^*) - \mathcal{L}(\boldsymbol{w}^*)$ from the optimization point of view where each random feature is taken as a "coordinate" in the infinite-dimensional space. This forms one contribution of this paper: extending the analysis of **?** to the case when $\ell(s, a)$ is smooth and strongly convex in a. Note even in this setting, $\mathcal{L}(\boldsymbol{w})$ is not strongly convex w.r.t. $\boldsymbol{w}$. However, here by leveraging the concept of Constant Nullspace Strong Convexity introduced in **?**, we show that the loss $\mathcal{L}(\boldsymbol{w}_d^*)$ given by d Random Features, interpreted as d steps of an infinite-dimensional (fully-corrective) randomized coordinate descent, has exponentially fast convergence to $\mathcal{L}(\boldsymbol{w}^*)$. We define the following two key assumptions:

Assumption 1. *The loss $\ell(s, a)$ is smooth w.r.t. the action a with parameter $\beta > 0$ iff its first derivative satisfies*

$$\|\nabla_a \ell(s, a) - \nabla_a \ell(s, a')\| \leq \beta \|a - a'\|.$$

Assumption 2. *The loss $\ell(s, a)$ is strongly convex w.r.t. the action a with parameter $m > 0$ iff*

$$\ell(s, a') - \ell(s, a) \geq \langle \nabla_a \ell(s, a), a\rangle + \frac{m}{2}\|a - a'\|^2.$$

The following lemma then shows that, for a strongly convex loss $\ell(s, a)$, the approximation error $\mathcal{L}(\boldsymbol{w}) - \mathcal{L}(\boldsymbol{w}^*)$ for some function $\boldsymbol{w}$ can be bounded by the square magnitude of gradient evaluated at point $\boldsymbol{w}$.

Lemma 1. *Let $\boldsymbol{w} \in \mathcal{H}_k$, and $\boldsymbol{w}^*$ be any reference function in the RKHS. We have*

$$\mathcal{L}(\boldsymbol{w}) - \mathcal{L}(\boldsymbol{w}^*) \leq \frac{1}{2\mu}\|\nabla \mathcal{L}(\boldsymbol{w})\|^2. \tag{6}$$

for some $\mu = m\lambda_k > 0$, where m is the strongly-convex parameter of loss $\ell(s, a)$ w.r.t. a and λ_k is the minimum positive eigenvalue of kernel matrix K where $K_{i,j} = \langle \boldsymbol{\phi}(s_i), \boldsymbol{\phi}(s_j)\rangle$.

Then by showing that the descent amount in loss $\mathcal{L}(\boldsymbol{w}_d)$ given by each new random feature added is proportional to the square magnitude of gradient $\nabla\mathcal{L}(\boldsymbol{w}_d)$, we are able to give the following theorem.

Theorem 2 (Approximation Error)**.** *Let $\mathcal{H}_k$ be the RKHS defined by the kernel $k(x, y)$, $\phi(x; \theta)$ be a feature map bounded in magnitude by $B \geq |\phi(x; \theta)|$, and $p(\theta)$ be a distribution such that Equation* (**??**) *holds. If loss $\ell(s, a)$ satisfies Assumptions* **??** *and* **??***, Then the expected approximation error for the optimal policy $\boldsymbol{w}_d^*$ in the space spanned by d random features is*

$$\mathbb{E}[\min_{\boldsymbol{w}\in\mathbb{R}^d} \mathcal{L}(\boldsymbol{w})] - \mathcal{L}(\boldsymbol{w}^*) \leq \gamma^d \left(\mathcal{L}(\boldsymbol{0}) - \mathcal{L}(\boldsymbol{w}^*)\right) \tag{7}$$

where $\gamma = 1 - \frac{\mu\lambda_k}{\beta B^2}$.

Table 2: Comparison of algorithms used in the experiments: Here b_i denotes the lower confidence bound, which is set to zero when selecting actions greedily. K denotes the number of discretized actions, n denotes the dimension of the raw state space s, and d denotes the dimension of the random feature representation $\phi(s)$.

Name	Action Set	Parameters	Policy
S-LinUCB	$[K]$	$\boldsymbol{w}_i \in \mathbb{R}^n, i \in [K]$	$\pi(s) = \min_i \boldsymbol{w}_i^\top s - \alpha b_i$
RF-LinUCB	$[K]$	$\boldsymbol{w}_i \in \mathbb{R}^d, i \in [K]$	$\pi(s) = \min_i \boldsymbol{w}_i^\top \phi(s) - \alpha b_i$
S-BGD	$\mathbb{R}$	$\boldsymbol{w} \in \mathbb{R}^n$	$\pi(s) = \boldsymbol{w}^\top s$
RF-BGD	$\mathbb{R}$	$\boldsymbol{w} \in \mathbb{R}^d$	$\pi(s) = \boldsymbol{w}^\top \phi(s)$

The proof is given in the appendix. The above theorem in *expectation* can be easily extended to the *high probability* result using Theorem 1 of **?**.

Theorem **??** provides a bound on the expected error of the best function in the set of functions spanned by the d random features. After the initial phase where the algorithm samples a finite number of features, it then performs a finite-dimensional bandit (zeroth-order) optimization. But the algorithm can only find an approximately optimal solution, resulting in estimation error. Therefore we use a finite dimensional estimation error result from **?** which utilizes the Bandit Gradient Descent (BGD) algorithm and assumes the loss function is smooth and strongly convex and the domain is unconstrained.

Theorem 3 (Optimization Error)**.** *From* **?** *Theorem 14* [1]*: Let $|\mathcal{L}_t(\boldsymbol{w})| \leq C$ for all $\boldsymbol{w}, t$. If $\mathcal{L}_t(\boldsymbol{w})$ is b-smooth and ν-strongly convex for all t and $\boldsymbol{w} \in \mathbb{R}^d$ is unconstrained, then the BGD algorithm with parameters $\delta = \left(\frac{d^2C^2(1+\log(T))}{3Tb\nu}\right)^{1/4}$ and $\eta_t = \frac{1}{t\nu}$ has an average regret bound*

$$\frac{1}{T}\sum_{t=1}^{T} \mathcal{L}_t(\boldsymbol{w}_t) - \min_{w \in \mathbb{R}^d} \mathcal{L}(\boldsymbol{w}) \leq dC\sqrt{\frac{3b(1+\log(T))}{\nu T}}.$$

We can combine the approximation error (Theorem **??**) and estimation error (Theorem **??**) bounds above to provide an $O(\sqrt{\frac{\log^3 T}{T}})$ average regret bound for the continuous-action nonparametric bandit problem, which is the primary theoretical contribution of this work.

Theorem 4 (Nonparametric Contextual Bandits)**.** *Let $r = \mathcal{L}(\mathbf{0}) - \mathcal{L}(\boldsymbol{w}^*)$ and $h = \frac{\nu r^2 \log^2(1/\gamma)}{3bC^2}$ and all other constants defined in the theorems above. Setting $d = \frac{1}{2\log(1/\gamma)} \log\left(\frac{hT}{1+\log T}\right)$ and using the schedules for δ_t and η_t given in Theorem* **??***, then the average regret of RF-BGD (Algorithm* **??***) is*

$$R(T) \leq r\sqrt{\frac{1+\log T}{hT}}\left(1 + \frac{1}{2}\log\left(\frac{hT}{1+\log T}\right)\right).$$

Proof. Combining Theorem **??** and Theorem **??** gives a bound for the average regret in terms of d and T. This bound can be minimized with respect to d , providing the value for d given above. Setting d to this minimizing value gives the stated $O(\sqrt{\frac{\log^3 T}{T}})$ result. □

4 Experiments

We empirically demonstrate the performance of Algorithm **??** on two benchmark problems. The first is a simple toy problem with a one-dimensional state which we refer to as the Planar Domain (as the state to action policy mapping can be visualized on a plane) and the second is the Half-Field Offense (HFO) soccer domain used in the reinforcement learning literature (**?**).

Our algorithm exploits some core properties of these domains: First, we assume that the action space is continuous. Second, we make the nonparametric assumption that the optimal policy belongs to an

[1] There is a minor error in the statement of this result in **?**: the RHS is divided by an extra factor of T

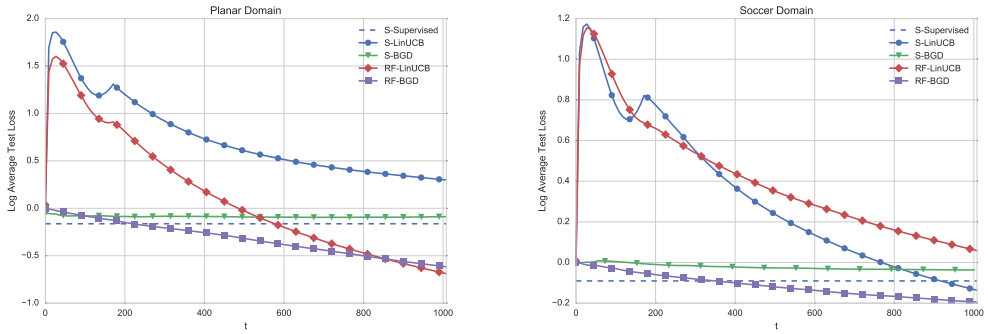

Figure 1: Comparison of the log average test loss for RF-BGD and the baselines in the Soccer and Planar domains

RKHS. Third, we assume that the loss is smooth and strongly convex with respect to the actions. As ours is the first algorithm that addresses all three of these assumptions simultaneously, we are forced to compare against algorithms designed for problems without at least one of these assumptions.

For our experiments we use the Gaussian kernel and Fourier features, as in **?**. As the Fourier transform of a Gaussian is also a Gaussian, the feature parameter distribution $p(\theta)$ is Gaussian. Further implementation details are given in the appendix.

We compare our method with those in Table **??**. Discrete action contextual bandits are very well-studied in the literature and a common approach when faced with a continuous action problem is to discretize the action space (e.g. divide the space into uniform intervals and "play" the center of the ith interval as discrete action/arm i). Discretizing the actions in this way treats each action independently, ignoring the ordinal structure relating the actions. This prevents the algorithm from using the structure of the action space to generalize knowledge about one action to other neighboring actions, resulting in an unnecessary degree of exploration and consequently slower learning. The discretization approach becomes infeasible in higher dimensional continuous action spaces as the number of actions needed grows exponentially with the dimension, while our algorithm does not explicitly depend on the dimension. Thus the one-dimensional action problems illustrated in these experiments are *generous* towards the discretized action baselines compared to problems with higher dimensional action spaces. Further, discretization-based approaches cannot be used when the action domain is unbounded, while our algorithm can easily be extended to that case.

We choose the LinUCB algorithm (**?**) for its simplicity and interpretability. LinUCB maintains a linear model of the mapping from state to expected cost for each discrete action, along with confidence bounds for each action quantifying the uncertainty resulting from limited experience with that action. In each round, LinUCB chooses the action with the highest expected reward (equivalently minimum loss) plus its upper confidence bound (equivalently minus its lower confidence bound). LinUCB requires a set of features for constructing its linear model. We compare against LinUCB using two different sets of features: the "raw" state representation s, and the random features used by our algorithm $\phi(s)$. We refer to these two variants as S-LinUCB and RF-LinUCB respectively.

The other prevalent approach in the contextual bandit literature is to assume that the optimal policy lies in a (finite-dimensional) parameterized policy class, typically linear in the state. These algorithms, while addressing the continuous-action setting directly, typically require domain knowledge to select an appropriate state space for the problem. If the provided state space is inadequate, these algorithms can perform very poorly. We therefore compare RF-BGD against BGD with a policy linear in the raw state representation s_t, denoted S-BGD. This baseline essentially uses the inner product kernel in the state space. Note that $d = n$ here (the ambient dimension of the state space), while for RF-BGD, d grows with T.

Figure **??** compares the statistical performance of RF-BGD with the baselines described above in terms of average (cumulative) test loss against the total number of function evaluations. We see that the LinUCB variants suffer through a costly exploration period. After this extensive exploration the LinUCB algorithms do perform quite well in terms of the final average regret–better than the BGD variants in some cases, which is indeed possible because the policies are different, see Table **??**–but

for most large-scale problems of interest, the user is typically in the low-sample regime where RF-BGD outperforms them.

Furthermore, it should be noted that we are showing the performance in terms of sample complexity. If we consider computational complexity, the UCB algorithms are very expensive as they solve a linear least-squares model at every step, costing $O(d^3)$ computation, while each RF-BGD step is $O(d)$. Thus the RF-BGD is the preferable algorithm both in the sample-constrained regime as well as the computation-constrained regime. Additionally, this difference in performance will grow quickly with the dimension of the action space because the number of discrete actions needed grows exponentially, as noted above. Also of note is that the LinUCB algorithms have K times as many parameters as the BGD equivalents (see Table **??**).

In Figure **??**, the dotted horizontal line represents the supervised baseline for the raw state policy parameterization, obtained by solving the oracle least squares regression problem, i.e. with access to the optimal actions, and represents the best performance that bandit gradient algorithms can achieve. In both domains, the supervised solution for the Random Feature parameterization lies below the scale of the plot. Finally, note that the Bandit Gradient policies differ from the supervised lower bound in the limit due to the constant sampling radius δ (in (**??**)) and learning rate η. In practice, decreasing schedules for δ increase instability and this small gap may be unavoidable. The LinUCB variants are not lower bounded by this supervised policy performance as the discrete actions provide a different policy parameterization.

5 Conclusion

After reviewing the bandit literature from an optimization perspective, we presented a novel non-parametric, continuous-action contextual bandit formulation and provided an optimization-based algorithm with an $O(\sqrt{\log^3(T)/T})$ average regret guarantee for losses that are smooth and strongly convex in the action. We then demonstrated the effectiveness of our approach empirically on two continuous action contextual bandit tasks, showing that our algorithm outperforms reasonable baselines while being very simple and computationally efficient.

References

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, 2011.

Alekh Agarwal, Ofer Dekel, and Lin Xiao. Optimal Algorithms for Online Convex Optimization with Multi-Point Bandit Feedback. In *COLT*, pages 28–40, 2010.

Alekh Agarwal, Dean P Foster, Daniel J Hsu, Sham M Kakade, and Alexander Rakhlin. Stochastic convex optimization with bandit feedback. In *Advances in Neural Information Processing Systems 24*, pages 1035–1043. 2011.

Alekh Agarwal, Daniel Hsu, Satyen Kale, John Langford, Lihong Li, and Robert E Schapire. Taming the monster: A fast and simple algorithm for contextual bandits. *arXiv preprint arXiv:1402.0555*, 2014.

Jean-Yves Audibert and Sébastien Bubeck. Minimax policies for adversarial and stochastic bandits. In *COLT*, pages 217–226, 2009.

Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The Nonstochastic Multi-armed Bandit Problem. *SIAM J. Comput.*, 32(1):48–77, 2002.

Dimitris Bertsimas and Santosh Vempala. Solving convex programs by random walks. *J. ACM*, 51 (4):540–556, Jul 2004. ISSN 0004-5411.

Alina Beygelzimer, John Langford, Lihong Li, Lev Reyzin, and Robert E Schapire. Contextual bandit algorithms with supervised learning guarantees. *arXiv preprint arXiv:1002.4058*, 2010.

Sébastien Bubeck, Rémi Munos, Gilles Stoltz, and Csaba Szepesvári. X-Armed Bandits. *J. Mach. Learn. Res.*, 12:1655–1695, Jul 2011.

Sébastien Bubeck, Nicolò Cesa-Bianchi, and Sham M Kakade. Towards minimax policies for online linear optimization with bandit feedback. 14 Feb 2012.

Wei Chu, Lihong Li, Lev Reyzin, and Robert E Schapire. Contextual bandits with linear payoff functions. In *International Conference on Artificial Intelligence and Statistics*, pages 208–214. machinelearning.wustl.edu, 2011.

Bo Dai, Bo Xie, Niao He, Yingyu Liang, Anant Raj, Maria-Florina F Balcan, and Le Song. Scalable Kernel Methods via Doubly Stochastic Gradients. In *Advances in Neural Information Processing Systems 27*, pages 3041–3049. 2014.

Marc Peter Deisenroth, Gerhard Neumann, Jan Peters, and Others. A Survey on Policy Search for Robotics. *Foundations and Trends in Robotics*, 2(1-2):1–142, 2013.

Abraham D Flaxman, Adam Tauman Kalai, and H Brendan McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 385–394. Society for Industrial and Applied Mathematics, 2005.

Elad Hazan, Alexander Rakhlin, and Peter L Bartlett. Adaptive online gradient descent. In *Advances in Neural Information Processing Systems*, pages 65–72. machinelearning.wustl.edu, 2007.

Matthias Hein and Olivier Bousquet. Kernels, associated structures and generalizations. *Max-Planck-Institut fuer biologische Kybernetik, Technical Report*, 2004.

Shivaram Kalyanakrishnan, Yaxin Liu, and Peter Stone. Half field offense in robocup soccer: A multiagent reinforcement learning case study. In *RoboCup 2006: Robot Soccer World Cup X*, pages 72–85. Springer, 2006.

Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. Bandits and Experts in Metric Spaces. 4 Dec 2013.

Robert D Kleinberg. Nearly tight bounds for the continuum-armed bandit problem. In *Advances in Neural Information Processing Systems*, pages 697–704. machinelearning.wustl.edu, 2004.

Huitian Lei, Ambuj Tewari, and Susan Murphy. An Actor-Critic Contextual Bandit Algorithm for Personalized Interventions using Mobile Devices. *Ann Arbor*, 1001:48109, 2014.

Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 661–670, New York, NY, USA, 2010. ACM. ISBN 9781605587998. doi: 10.1145/1772690.1772758.

Michael L Littman. Reinforcement learning improves behaviour from evaluative feedback. *Nature*, 521(7553):445–451, 28 May 2015. ISSN 0028-0836, 1476-4687.

H Brendan McMahan and Matthew J Streeter. Tighter Bounds for Multi-Armed Bandits with Expert Advice. In *COLT*, 2009.

Arkadi Nemirovsky. Problem complexity and method efficiency in optimization. 1983.

Yurii Nesterov and Others. Random gradient-free minimization of convex functions. Technical report, Université catholique de Louvain, Center for Operations Research and Econometrics (CORE), 2011.

Andrew Y Ng and Michael Jordan. PEGASUS: A Policy Search Method for Large MDPs and POMDPs. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, UAI'00, pages 406–415, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

Ali Rahimi and Benjamin Recht. Weighted Sums of Random Kitchen Sinks: Replacing minimization with randomization in learning. In D Koller, D Schuurmans, Y Bengio, and L Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1313–1320. Curran Associates, Inc., 2009.

Alexander Rakhlin, Ohad Shamir, and Karthik Sridharan. Making Gradient Descent Optimal for Strongly Convex Stochastic Optimization. 26 Sep 2011.

Peter Richtárik and Martin Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Math. Program.*, 144(1-2):1–38, 2014.

Ohad Shamir. On the Complexity of Bandit and Derivative-Free Stochastic Convex Optimization. 11 Sep 2012.

Aleksandrs Slivkins. Contextual bandits with similarity information. *The Journal of Machine Learning Research*, 15(1):2533–2568, 2014.

Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design. 21 Dec 2009.

Richard S Sutton, David A McAllester, Satinder P Singh, Yishay Mansour, and Others. Policy Gradient Methods for Reinforcement Learning with Function Approximation. In *NIPS*, volume 99, pages 1057–1063. Citeseer, 1999.

Michal Valko, Nathaniel Korda, Remi Munos, Ilias Flaounas, and Nelo Cristianini. Finite-Time Analysis of Kernelised Contextual Bandits. 26 Sep 2013.

Chih-Chun Wang, S R Kulkarni, and H V Poor. Bandit problems with side observations. *IEEE Trans. Automat. Contr.*, 50(3):338–355, Mar 2005. ISSN 0018-9286.

Ian En-Hsu Yen, Cho-Jui Hsieh, Pradeep K Ravikumar, and Inderjit S Dhillon. Constant Nullspace Strong Convexity and Fast Convergence of Proximal Methods under High-Dimensional Settings. In *Advances in Neural Information Processing Systems 27*, pages 1008–1016. 2014a.

Ian En-Hsu Yen, Ting-Wei Lin, Shou-De Lin, Pradeep K Ravikumar, and Inderjit S Dhillon. Sparse Random Feature Algorithm as Coordinate Descent in Hilbert Space. In *Advances in Neural Information Processing Systems 27*, pages 2456–2464. 2014b.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. 2003.

Appendix

A Proof for Lemma 1

Proof. Let $\mathcal{N}$ be the Nullspace given by $\Phi = \{\boldsymbol{\phi}(\boldsymbol{s}_t)|t \in [T]\}$ which satifies

$$\Phi \boldsymbol{w} = \sum_{t=1}^{T} \langle \boldsymbol{\phi}(\boldsymbol{s}_t), \boldsymbol{w} \rangle = 0, \ \forall \boldsymbol{w} \in \mathcal{N}, \tag{8}$$

and denote $\mathcal{N}^\perp$ as the orthogonal space of $\mathcal{N}$. Then decomposing $\boldsymbol{w}, \boldsymbol{w}^*$ as $\boldsymbol{w} = \boldsymbol{u}+\boldsymbol{v}, \boldsymbol{w}^* = \boldsymbol{u}^*+\boldsymbol{v}^*$ where $\boldsymbol{u}, \boldsymbol{u}^* \in \mathcal{N}$, $\boldsymbol{v}, \boldsymbol{v}^* \in \mathcal{N}^\perp$. We have

$$\mathcal{L}(\boldsymbol{v}) = \mathcal{L}(\boldsymbol{w}) \ , \ \mathcal{L}(\boldsymbol{v}^*) = \mathcal{L}(\boldsymbol{w}^*) \ , \ \nabla \mathcal{L}(\boldsymbol{w}) = \nabla \mathcal{L}(\boldsymbol{v}).$$

since the components in the Nullspace $\mathcal{N}$ does not contribute to any difference to the change in objective from its definition (**??**). Therefore, for any $\Delta \boldsymbol{v} \in \mathcal{N}^\perp$, we have

$$\begin{aligned}
\mathcal{L}(\boldsymbol{v} + \Delta \boldsymbol{v}) - \mathcal{L}(\boldsymbol{v}) &\geq \langle \nabla \mathcal{L}(\boldsymbol{v}), \Delta \boldsymbol{v} \rangle + \frac{m}{2} \|\Phi \Delta \boldsymbol{v}\|^2 \\
&\geq \langle \nabla \mathcal{L}(\boldsymbol{v}), \Delta \boldsymbol{v} \rangle + \frac{m \lambda_k}{2} \|\Delta \boldsymbol{v}\|^2 \\
&\geq -\frac{1}{2\mu} \|\nabla \mathcal{L}(\boldsymbol{v})\|^2,
\end{aligned}$$

where the firat inequality is from the strong convexity of ℓ, the second inequality holds because $\boldsymbol{v} \in \mathcal{N}^\perp$, and the third inequality is obtained from minimizing the RHS. Then we reach our result

$$\begin{aligned}
\mathcal{L}(\boldsymbol{w}^*) - \mathcal{L}(\boldsymbol{w}) &= \mathcal{L}(\boldsymbol{v}^*) - \mathcal{L}(\boldsymbol{v}) \\
&\geq -\frac{1}{2\mu} \|\nabla \mathcal{L}(\boldsymbol{v})\|^2 \\
&= -\frac{1}{2\mu} \|\nabla \mathcal{L}(\boldsymbol{w})\|^2,
\end{aligned}$$

which leads to the conclusion. □

B Proof of Theorem 2

Proof. We interpret Random Features as Randomized Coordinate Descent with coordinates drawn from distribution $p(\theta)$ that minimizes the objective

$$G(\bar{\boldsymbol{w}}) = \sum_{t=1}^{T} \ell(s_t, \langle \bar{\boldsymbol{w}}, \bar{\boldsymbol{\phi}}(s_t) \rangle) \tag{9}$$

where $\boldsymbol{\phi}(.) = \sqrt{\boldsymbol{p}} \circ \bar{\boldsymbol{\phi}}(.)$. Since $\ell(s, a)$ is smooth with parameter β in a, the minimization w.r.t. a coordinate θ has

$$\begin{aligned}
&\ell(\langle \bar{\boldsymbol{w}} + \eta \boldsymbol{\delta}_h, \bar{\boldsymbol{\phi}}(s) \rangle) - \ell(s, \langle \bar{\boldsymbol{w}}, \bar{\boldsymbol{\phi}}(s) \rangle) \\
&\leq \nabla \ell(s, \langle \bar{\boldsymbol{w}}, \bar{\boldsymbol{\phi}}(s) \rangle) \eta \bar{\phi}(s; \theta) + \frac{\beta B^2}{2} \eta^2.
\end{aligned}$$

where B is an upper bound on $|\bar{\phi}(s, \theta)|$. Taking empirical sum over the T rounds, we have

$$G(\bar{\boldsymbol{w}} + \eta \boldsymbol{\delta}_h) - G(\bar{\boldsymbol{w}}) \leq g_\theta \eta + \frac{\beta B^2}{2} \eta^2,$$

where

$$g_\theta = \sum_{t=1}^{T} \nabla \ell(s_t, \langle \bar{\boldsymbol{w}}, \bar{\boldsymbol{\phi}}(s_t) \rangle) \bar{\phi}(s_t; \theta).$$

Taking minimizer of both sides w.r.t. η results in

$$G(\bar{\boldsymbol{w}} + \eta^* \boldsymbol{\delta}_h) - G(\bar{\boldsymbol{w}}) \leq -\frac{g_h^2}{2\beta B^2},$$

and the taking expectation w.r.t. the distribution $p(\theta)$ from which coordinate is drawn, we have

$$\begin{aligned} &\mathbb{E}_\theta[G(\bar{\boldsymbol{w}} + \eta^* \boldsymbol{\delta}_\theta)] - G(\bar{\boldsymbol{w}}) \\ &\leq \frac{1}{2\beta B^2} \int_\theta p(\theta) g_\theta^2 d\theta = \frac{1}{2\beta B^2} \|\nabla \mathcal{L}(\boldsymbol{w})\|^2, \end{aligned} \tag{10}$$

where the last equality follows directly by the definition of $\nabla \mathcal{L}(\boldsymbol{w})$ evaluated at $\boldsymbol{w}$, where $\bar{\boldsymbol{w}} = \sqrt{\boldsymbol{p}} \circ \boldsymbol{w}$.

Now notice that $G(\bar{\boldsymbol{w}}^t) = \mathcal{L}(\boldsymbol{w}^t)$, and by Lemma (**??**) and (**??**), we have

$$\begin{aligned} &\mathbb{E}_{\theta^{(t+1)}}[\mathcal{L}(\boldsymbol{w}^{(t+1)})] - \mathcal{L}(\boldsymbol{w}^{(t)}) \\ &\leq -\frac{\|\nabla \mathcal{L}(\boldsymbol{w}^{(t)})\|^2}{2\beta B^2} \leq -\frac{\mu(\mathcal{L}(\boldsymbol{w}^{(t)}) - \mathcal{L}(\boldsymbol{w}^*))}{\beta B^2} \end{aligned} \tag{11}$$

for any reference $\boldsymbol{w}^*$ in the RKHS. Define $\Delta^t = \mathbb{E}_{\theta^{(t)}}[\mathcal{L}(\boldsymbol{w}^{(t)})] - \mathcal{L}(\boldsymbol{w}^*)$. Then taking expectation over $\theta^{(t)}$ on the inequality (**??**), we have

$$\Delta^{t+1} - \Delta^t \leq -\frac{\mu}{\beta B^2} \Delta^t.$$

Recursively applying the above inequality leads to the conclusion. □

C Proof of Theorem 4

Proof. Combining theorem 2 and 3, the average regret can be expressed as sum of the estimation error and the approximation error:

$$R_d(T) = \frac{1}{T} \sum_{t=1}^{T} \mathcal{L}_t(\boldsymbol{w}_t) - \mathcal{L}_t(\boldsymbol{w}^*) = \Delta \gamma^d + dA$$

where $\Delta = \mathcal{L}(0) - \mathcal{L}(\boldsymbol{w}^*)$ and $A = \sqrt{\frac{3bC^2(1+\log T)}{mT}}$. Taking the derivative wrt d and setting it to zero we get

$$\Delta \log(\gamma) e^{d \log(\gamma)} + A = 0$$

solving for d gives

$$d == \frac{1}{2\log(1/\gamma)} \log\left(\frac{hT}{1+\log T}\right)$$

where $h = \frac{m\Delta^2 \log^2(1/\gamma)}{3bC^2}$. Using this value for d, the regret is then restated as

$$R(T) = \Delta \sqrt{\frac{1+\log T}{hT}} \left(1 + \frac{1}{2} \log\left(\frac{hT}{1+\log T}\right)\right)$$

which is $O(\sqrt{\frac{\log^3 T}{T}})$.

□

D Experimental Details

Implementation Details The hyperparameters for both BGD and LinUCB were optimized for each domain. For LinUCB this includes the number of discrete actions K and the exploration parameter α which controls the relative weight of the confidence bounds versus the empirical expected loss during action selection (see table **??**). For BGD, a schedule for the learning rate and the sampling radius δ must be chosen. We found that constant values for these were sufficient for our purposes, despite the theory which suggests decreasing schedules. It was also useful to use multiple function evaluations and sampling directions when estimating the gradient; in our experiments we averaged over three samples of the unit direction vectors u and three function evaluations $\ell_t(\boldsymbol{w} + \delta u)$ for each u, costing 9 total function evaluations per gradient step. To provide a fair comparison, the plots given in figure **??** are against the total number of function evaluations t, not gradient iterations. To further decrease variance, we subtracted the average function value from the previous gradient iteration from the current function value when estimating the gradient. Thus the gradient update direction is $g_t = u(f_t - f_{t-1})$; note that it is not necessary to multiply by the d/δ term in Equation (**??**) as this scalar can be absorbed into the learning rate (see Algorithm **??**).

In the experiments, we used $d = 100$, $T = 1000$, unit variance for $p(\theta)$, $\alpha = 1$, $K = 20$, $\eta = 0.1$, and $\delta = 1$ in both domains.

Domain Details The Planar Domain consists of a one-dimensional state with an optimal policy of $\pi^*(s) = 2s^3 - s$ and the loss being the squared difference from this optimal policy: $\ell(s, a) = (\pi^*(s) - a)^2$. The Soccer Domain has a two-dimensional state representing the agent's position on the field and the action space is the angle to kick the ball in order to score a goal. The loss is the squared difference between the action chosen and the angle to the goal center.