Subspace Learning with Gaussian Processes for Sparse Contextual Bandits

Yair Chizi[1]

Supervisor: Julia Olkhovskaia[1]

[1]**EEMCS, Delft University of Technology, The Netherlands**

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 22, 2025

Name of the student: Yair Chizi
Final project course: CSE3000 Research Project
Thesis committee: Julia Olkhovskaia, Luciano C. Siebert

Abstract

The multi-armed bandit problem is a sequential learning scenario in which a learning algorithm seeks to obtain rewards by selecting an arm, or action, in each round, given limited initial knowledge. Contextual bandits present an additional context every round that informs the bandit algorithm and guides decision-making. While successfully applied in practice, research continues to explore efficient bandit algorithms for high-dimensional bandits with nonparametric, sparsely varying reward functions. One such algorithm is the two-phase SI-BO algorithm, which incorporates an initial subspace learning phase to identify the effective context subspace on which the function varies, and a subsequent Bayesian optimization phase that applies the Gaussian Process-based GP-UCB algorithm to the learned subspace. While the SI-BO offers a theoretical regret performance with weak sub-exponential dependence on the ambient dimension, it is hindered by a high computational cost stemming from the Gaussian Process regression. Building on the algorithm framework introduced in SI-BO, this paper aims to investigate the empirical regret performance of Gaussian Process-based learning algorithms that incorporate subspace learning. To that end, we introduce a novel algorithm, SI-BKB, which combines the subspace learning in SI-BO with the BKB sketching algorithm, reducing computational complexity while maintaining theoretical guarantees. Through synthetic data generation, this paper presents a systematic empirical study on linear and nonlinear bandit environments with varying levels of sparsity. The results demonstrate that the SI-BKB algorithm has comparable regret performance to the SI-BO. Additionally, the regret performance indicates that misalignment of the learned subspace results in suboptimal regret performance during the optimization phase. Moreover, we demonstrate that high sparsity, through subspace misalignment, can improve the regret performance. Repository is available at https://github.com/Cheese-1/SparseSequentialLearning.

1 Introduction

The multi-armed bandit problem is a sequential learning scenario in which a learning algorithm seeks to obtain rewards by selecting an arm, or action, in each round, given limited initial knowledge. Extending the problem, contextual bandits present an additional context every round that is leveraged by the bandit algorithm to guide decision-making. Succinctly, the iterative interaction protocol of contextual bandits involves the bandit algorithms observing the current context, selecting an available action, and then receiving a corresponding reward from the environment [7].

In theoretical settings, performance is typically evaluated relative to the best (constant-action) algorithm through cumulative regret, which is the difference in cumulative expected reward between the learner algorithm and the best (constant-action) algorithm [7]. The theoretical performance of bandit algorithms is often evaluated by the dependence of the asymptotic regret upper bound on the horizon T—the total number of rounds; however, in high-dimensional settings, it is also customary to consider the dependence on the ambient dimension of the context d.

Variants of contextual bandit algorithms have found success in diverse application domains. These include personalized news recommendation systems that adaptively model user preferences [8], online network routing in adversarial environments [1], and self-financing portfolio selection over risky assets [14]. More recently, they have also been applied to high-stakes problems in personalized medicine, such as dosing strategies for Warfarin based on patient-specific features [2].

In many applications, the provided context information is high-dimensional but only weakly informative. In other words, practical applications can have a low-dimensional reward function that is embedded in the high-dimensional ambient space. These scenarios are characterized as sparse contextual bandits.

The effect of high-dimensionality is most apparent in the regret bounds of bandit algorithms. In specialized, well-studied bandits, regret bounds attain a polynomial dependence on the ambient dimension d. For example, in linear contextual bandits, the `LinUCB` bandit algorithm achieves linear dependence, with an upper regret bound of $O(d\sqrt{T}\log T)$[7; 8]. In contrast, nonparametric contextual bandits, where the reward function is scantily modeled, often suffer from regret bounds that depend exponentially on the ambient dimension d, rendering them intractable in high-dimensional regimes [9; 12; 15].

One bandit algorithm for nonparametric sparse functions that achieves sub-exponential dependence on d is the `SI-BO` algorithm [5]. `SI-BO` operates in two phases: the first sacrifices early rounds to learning the underlying subspace, while the second applies the Gaussian Process-based `GP-UCB` algorithm on the learned subspace. The two phases directly correspond to the exploration-exploitation tradeoff that is central to bandit problems; early exploration allows for the refinement of a well-aligned subspace, but incurs regret penalties by delaying the actual application of the bandit algorithm.

While `SI-BO` offers improvements to the regret bounds, in theory, computational challenges should impede its effectiveness. In particular, the inclusion of the `GP-UCB` algorithm necessitates the inversion of a matrix whose size directly scales with the horizon, incurring substantial space and time computational costs [3]. Due to this reason, the `SI-BO` is unsuitable for bandit problems with a large horizon, ultimately limiting the duration of possible subspace learning altogether.

While the theoretical analysis of `SI-BO` is well-established in [5], an empirical evaluation of its framework is yet to be explored in practice. Hence, this paper aims to investigate an empirical study of subspace learning on Gaussian Processes-based learning algorithms under sparse conditions. To that end, we investigate the regret performance of `SI-BO` and a novel algorithm, `SI-BKB`, in linear and non-linear bandits

with varying sparsity. The empirical investigation focuses on three key questions: (i) how can the framework of SI-BO be extended to create the SI-BKB algorithm while maintaining the theoretical regret performance; (ii) does SI-BKB offer competitive performance relative to SI-BO following subspace identification; and (iii) what trade-offs emerge between the initial subspace identification cost and long-term optimization benefit.

To address these research questions, the paper is structured to begin with the theoretical foundations and conclude with the empirical analysis. Section 2 provides a formal definition for sparse contextual bandits and the regret metric. Section 3 formalizes the GP-UCB, BKB, and SI-BO algorithms that appear in prior works. Section 4 introduces the novel SI-BKB algorithm and derives the theoretical regret bounds. Section 5 outlines the experimental setup for the synthetic data generation. Section 6 exhibits the regret performance of the SI-BO and SI-BKB algorithms in linear and nonlinear bandits. Section 7 discusses the effect of misalignment and high sparsity on empirical regret. Section 9 reflects on the reproducibility and external impact of the research.

For ease of readability, the notation employed in this paper is available in Appendix A.

2 Problem Description

In this paper, we adopt a modified definition of contextual multi-armed bandits with an interaction protocol specified in [7]. At every round, a bandit algorithm must select a single arm, or action, $a \in \mathcal{A}$ with an associated context. Assuming a finite horizon T, at each round $t \in [T]$, the environment samples a context vector $x_{t,a} \sim \mathcal{D}_a$ for every arm from a distribution over a subset of the ambient space $\mathcal{X} \subseteq \mathbb{R}^d$. Subsequently, informed by the contexts, the bandit algorithm selects an action $a_t \in \mathcal{A}$ to observe a reward through a noisy oracle

$$y_t = f(x_{t,a_t}) + \eta_t \tag{1}$$

where $f : \mathbb{R}^d \to \mathbb{R}$ represents the environment's reward function and η_t is an i.i.d. sample of some Gaussian noise with known variance σ^2.

To incorporate sparsity, we further suppose that the reward function effectively varies only over a low-dimension subspace (*see* [5]). Formally, we assume that the reward function comprises of a reward function $g : \mathbb{R}^k \to \mathbb{R}$ function in the subspace and a linear transformation $A \in \mathbb{R}^{k \times d}$ with orthogonal rows so that

$$f(x) = g(Ax) \tag{2}$$

Notably, the generalized definition of sparsity specified in Eq. 2 encompasses the well-explored sparse linear bandit problem.. Linear bandits restrict the noisy oracle in Eq. 1 by requiring that f is an inner product with some unknown parameter vector $\theta_* \in \mathbb{R}^d$ so that [1]

$$y_{t,\text{linear}} = \langle x_{t,a_t}, \theta_* \rangle + \eta_t. \tag{3}$$

[1]In this case $x_{t,a'}$ refers to the feature vector provided by some feature selection map ϕ over the context and the action. Conventionally, the feature selection map is included in the definition of the oracle; however, due to irrelevancy, it is omitted here.

In the sparse variant, the parameter vector possesses a small support $|\text{Supp}(\theta_*)| = k < d$. Therefore, by considering the standard basis $e_1, \ldots, e_d$ in $\mathbb{R}^d$, we may represent sparse linear bandits in the form of Eq. 2 by specifying that

$$A_{\text{linear}} = [e_{o_1} \ldots e_{o_k}]^T \text{ where } o_1, \ldots, o_k \in \text{Supp}(\theta_*)$$
$$g(A_{\text{linear}} x) = \langle A_{\text{linear}} x, A_{\text{linear}} \theta_* \rangle + \eta_t$$

To ensure the validity of the bandit algorithms presented in Section 3, we impose further restrictions over the oracle and reward function in theory. First, we suppose that the oracle defined in Eq. 1 may be invoked at any time, with any context vector input, to produce an immediate reward. Second, we maintain that g is twice-differentiable with Lipschitz-continuous 2^{nd}-order derivatives and possesses a full-rank Hessian at $x = 0$. Finally, we presume that g lies in some Reproducible Kernel Hilbert Space (RKHS) over functions. Notably, the second restriction is violated for linear bandits since the Hessian for all multivariate linear functions is a zero matrix, and thus is rank zero.

2.1 Performance Objective

Although it is natural to assess performance based on total reward, bandit algorithms are typically evaluated in comparison to other algorithms. Conventionally, the learner's performance is measured in terms of cumulative regret with respect to the best constant-action algorithm in hindsight. Defining the random variable X_t to denote the reward obtainable by the learner in round t, the cumulative regret after T rounds is given by

$$R_T = T \max_{a \in \mathcal{A}} \mathbb{E}[f(x_{a,t})] - \mathbb{E}\left[\sum_{t=1}^{T} X_t\right]$$

The objective of the learner is to minimize this cumulative regret over time, effectively learning to select actions whose expected rewards approach those of the best fixed action algorithm. Hence, for an infinite horizon, the average cumulative regret should approach zero.

$$\lim_{T \to \infty} R_T / T = 0$$

3 Prior Work: Bandit Algorithms

In this paper, we evaluate the impact of subspace learning on Bayesian bandit learning algorithms that rely on Gaussian Process Regression. To that end, we consider two-phased algorithms consisting of a subspace learning phase and a Bayesian Optimization phase, commensurate with the approach laid by [5]. Sections 3.1 and 3.2 present two Gaussian Process-based bandit algorithms. Section 3.3 outlines the SI-BO algorithm's approach to subspace learning.

3.1 Bayesian Optimization for Gaussian Processes: GP-UCB

Upper Confidence Bound (UCB) algorithms form a class of common and well-studied learning algorithms. UCB algorithms operate under the principle of Optimism in the Face of Uncertainty, by greedily selecting actions that maximize an overestimate of the reward. Formally, in every round $t \in [T]$,

UCB algorithms endeavor to select an arm that maximizes some time-dependent index function as follows

$$a_t = \arg\max_{a \in \mathcal{A}} \mathrm{UCB}_t(x_{t,a}) \tag{4}$$

The UCB index function typically combines the expected reward for the given arm and the associated uncertainty computed over previously-observed oracle queries (*see* UCB, LinUCB in [7]). Such construction encourages early exploration of available arms due to the inherent initial high uncertainty.

GP-UCB is one variant of UCB algorithms that incorporates the predictive Bayesian uncertainty, under the assumption of Gaussian Processes priors, in the index function. The algorithm for GP-UCB is outlined procedurally in Algorithm 3 (Appendix B). The upper bounds on regret depend largely on the choice of kernel; for instance, the RBF kernel yields the regret bound $R_{T,\mathrm{RBF}} = \mathcal{O}(\sqrt{T(\log T)}^{d+1})$ that depends exponentially on the ambient dimension d [15].

Gaussian Processes are an assumption used in Bayesian regression tasks that yields an analytic computation of the mean and variance of the posterior predictive distribution. A Gaussian Process $f(x) \sim GP(0, \kappa(x,x))$[2] is a stochastic process over the context space $\mathcal{X}$ such that $(f(x_1), \ldots, f(x_t))$ is a multivariate normal distribution for any choice of inputs $X^* = \{x_j\}_{j=1}^t$. In accordance with the noisy oracle defined in Eq. 1, the probability of observing rewards $y = [y_1, \ldots, y_t]^T$ given observations X^* and a reward function f is normally distributed $\mathcal{N}(f(x), \sigma^2)$. Using Bayesian inference, the posterior probability is provided by

$$p(f \mid y, X^*) \propto p(f|X^*)p(y|f, X^*)$$

which suggests that the posterior distribution is also normally distributed. Given the past t contexts X_t^*, the mean and variance of the posterior distribution may be computed analytically [16] as follows

$$\begin{aligned} \mu_t(x) &= k_t(x)^T (K(X_t^*) + \sigma^2 I_t)^{-1} y \\ \sigma_t^2(x) &= \kappa(x,x) - k_t(x)^T (K(X_t^*) + \sigma^2 I_t)^{-1} k_t(x) \end{aligned} \tag{5}$$

where $K(X_t^*)$ is the covariance matrix over the contexts X_t^* and $k_t(x) = [\kappa(x, x_1) \ldots \kappa(x, x_t)]^T$ is the vector of the covariance of x with the contexts in X_t^*.

Using the regressor in Eq. 5, GP-UCB defines the index function using the posterior mean and variance. Using the posterior mean and uncertainty, we construct the following index function [16].

$$\mathrm{UCB_t}(x) = \mu_t(x) + \beta_t^{1/2} \sigma_t(x) \tag{6}$$

The index function in Eq. 6 modulates the uncertainty expressed by the posterior uncertainty through an exploration factor $\beta_t = 2B + 300\gamma_t \log^3(t/\delta)$ where $0 < \delta < 1$ is the desired accuracy rate, B is some upper bound on the complexity of the reward function $||f||_\kappa^2 < B$, and γ_t is the current maximal information gain.

[2] As demonstrated in [11], the mean function of the prior distribution, unconditioned by any observations, may be zero.

Isotropic Kernels

The choice of the underlying kernel in the Gaussian Process determines the characteristics of the function it best approximates. A kernel $\kappa : \mathbb{R}^d \times R^d \to \mathbb{R}$ is a symmetric positive definite function. In Reproducible Kernel Hilbert Spaces over functions where the kernel induces the inner product, every function may be defined in terms of the kernel $f(\cdot) = \sum_{j=1}^m \alpha_j \kappa(\cdot, x_j)$, where $\{x_j\}_{j=1}^m$ is some representation of points. In this paper, we are interested in three isotropic kernels, which are kernels that depend on the Euclidean distance of the inputs $\tau = ||x - x'||_2$ [13]. First, the Radial Basis Function (RBF) kernel is one such kernel that is infinitely smooth. The RBF kernel is parametrized by a length-scale parameter ℓ that dilates the distance. Second, the Matérn kernel is a generalization of the RBF kernel that controls the smoothness characteristic $\nu > 0$. As the smoothness characteristic approaches infinity, the Matérn kernel approximates the RBF kernel. Third, the Rational Quadratic (RQ) kernel is a mixture of RBF kernels at varying length scales, parametrized by a global length scale ℓ and a variance weight α. Formally, the three kernels are defined as

$$\begin{aligned} \kappa_{\mathrm{RBF}}(x, x') &= \exp\big(\frac{\tau^2}{2\ell^2}\big) \\ \kappa_{\mathrm{RQ}}(x, x') &= \big(1 + \frac{\tau^2}{2\alpha\ell^2}\big)^{-\alpha} \\ \kappa_{\mathrm{Matérn}}(x, x') &= \frac{2^{1-\nu}}{\Gamma(\nu)} \big(\frac{\sqrt{2\nu}\tau}{\ell}\big)^{\nu} K_\nu\big(\frac{\sqrt{2\nu}\tau}{\ell}\big) \end{aligned} \tag{7}$$

where $\Gamma(\cdot)$ is the Gamma Function, and $K_\nu(\cdot)$ is the modified Bessel function of the second kind. While the general Matérn kernel employs the computationally expensive Bessel function, relatively inexpensive closed-form formulations exist for select smoothness characteristics $\nu \in \{1/2, 3/2, 5/2\}$.

3.2 Sketching Gaussian Processes: BKB

While kernels provide an analytical framework for computing the parameters of the posterior predictive distribution, the efficiency of the computation is significantly hindered by the number of observed oracle queries. Naïvely, the GP-UCB algorithm computes the posterior mean and the posterior over the entire set of oracle queries in Eq. 5— a set that expands every round. As such, the size of the inverted matrix $K(X_t^*) + \sigma^2 I_t$ scales directly with the number of rounds. Inverting this square positive-definite symmetric matrix with size $t \times t$ incurs an exceptionally expensive cubic computational cost $O(t^3)$ [3].

As a remedy, the Budgeted Kernel Bandit (BKB) algorithm proposes to approximate the GP-UCB's index function by constructing a subset $S_t \subseteq \mathcal{D}$ of "inducing points" [3] that is refined every round. By considering a limited subset of points so that $|S_t| = m_t \leq t$, irrelevant, impotent queries are ignored, reducing unnecessary computation. The BKB algorithm is outlined in Algorithm 1. Similarly to GP-UCB, the BKB algorithm achieves a sublinear upper bound on regret, $R_T = \mathcal{O}(\sqrt{T} \log T)$ [3].

Using the inducing subset, BKB transforms context vectors into Nyström embeddings $z_t : \mathbb{R}^d \to \mathbb{R}^{m_t}$ as follows [3]

$$z_t(x) = \left(\mathbf{K}_{S_t}^{1/2}\right)^+ k_{S_t}(x) \tag{8}$$

where $\mathbf{K}_{S_t}$ is the covariance matrix of the inducing subset concerning the kernel, $(\cdot)^+$ is the pseudo-inverse operator, and $k_{S_t}(x_i) = [\kappa(S_{t,1}, x_i), \ldots, \kappa(S_{t,m_t}, x_i)]^T$ is the pairwise covariance vector. BKB then approximates the posterior mean and variance required for the GP-UCB index function in Eq. 6 as follows

$$\begin{aligned} \tilde{\mu}_t(x) &= z_t(x)^T V_t^{-1} Z_t(X_t)^T y_t \\ \tilde{\sigma}_t^2(x) &= \frac{1}{\sigma^2}\left(\kappa(x,x) - z_t(x)^T Z_t(X_t)^T V_t^{-1} z_t(x)\right) \end{aligned} \tag{9}$$

where $Z_t(X_t) = [z_t(x_1) \ldots z_t(x_t)]^T \in \mathbb{R}^{t \times m}$ is matrix of Nyström embeddings associated with S_t, and $V_t = Z_t(X_t)^T Z_t(X_t) + \sigma^2 I_t \in \mathbb{R}^{m \times m}$. In this approximation, the inverted matrix V_t no longer directly depends on the number of queries and is relatively smaller in size.

Notably, the approximations given in Eq. 9 do not directly correspond to Gaussian Process posteriors. By computing the kernel $k(x,x)$ in $\tilde{\sigma}_t(x)$, the uncertainty estimate avoids variance starvation that is likely to occur when using $z_t(x)^T z_t(x)$ (cf. [10]).

At every round, the set of inducing points is generated probabilistically. The probability of inclusion of a given past context is proportional to its uncertainty in the current Nyström embedding. Since the uncertainty negatively correlates with the number of contexts lying in some neighborhood, this probabilistic approach extracts only a subset of such contexts without largely impacting the uncertainty [3].

Algorithm 1 The BKB Algorithm adapted from [3]

Require: kernel function κ, B, δ, σ, $\bar{q}$, horizon n

Select an action $a_1 \in \mathcal{A}$ uniformly and observe the reward y_1.
Initialize the dataset $\mathcal{D} = \{(x_{1,a_1}, y_1)\}$ and the inducing subset $S_1 = \{x_{1,a_1}\}$.
for $t = 2$ to T **do**
  Observe the context $x_{t,a}$ for every action $a \in \mathcal{A}$.
  Compute the mean $\tilde{\mu}_t$ and variance $\tilde{\sigma}_t^2$ for every context vector using Eq. 9.
  Compute $\beta_t = 2B + 300\gamma_t \log^3(t/\delta)$
  Select $a_t = \arg\max_{a \in \mathcal{A}} .\tilde{\mu}_t(x_{t,a_t}) + \beta_t^{1/2}\tilde{\sigma}_t(x_{t,a_t})$
  Set $x_t = x_{t,a_t}$ and observe y_t.
  $\mathcal{D} \leftarrow \mathcal{D} \cup \{(x_{t,a_t}, y_t)\}$.
  for $i = 1$ to t **do**
    Set the probability $\tilde{p}_{t,i} = \bar{q} \cdot \tilde{\sigma}_t^2(\tilde{x}_i)$ where $\tilde{x}_i$ is the context observed in round i.
    Sample $q_{t,i} \sim \mathrm{Ber}(\tilde{p}_{t,i})$.
    if $q_{t,i} = 1$ **then** include $\tilde{x}_i$ in S_t.
  end for
end for

3.3 Subspace Learning: SI-BO

The SI-BO algorithm [5] incorporates subspace learning to apply the GP-UCB algorithm on a low-dimensional subspace. The algorithm operates in two phases: a Subspace Identification (SI) phase that attempts to approximate the transformation matrix A in Eq. 2; and a Bayesian Optimization (BO) phase that applies the GP-UCB learning algorithm directly on the learned subspace. The SI-BO algorithm is outlined in Algorithm 4 (Appendix B).

An advantage of SI-BO is that it offers a sub-exponential dependency of the upper regret bound on the ambient dimension. For both noisy and noiseless oracles, the regret is bounded by a polynomial ambient dimension term and the regret incurred by the GP-UCB. Since GP-UCB operates on the subspace, its regret is bounded exponentially on the subspace dimension k rather than the ambient dimension d. For this reason, the polynomial term dominates the GP-UCB regret in the ambient dimension, resulting in polynomial bounds.

In SI-BO, the subspace learning is facilitated through low-rank matrix reconstruction. In particular, we are interested in determining a low-rank approximation for the gradient matrix $X = [\nabla f(x_1) \ldots \nabla f(x_{m_X})]^T$ for m_X context samples [6]. To approximate X, we apply multiple linearization approximations with the step size ε for the directional derivative. In particular, if we sample m_φ matrices Φ_i of m_X directional vectors $\Phi_{i,j}$ uniformly sampled from $\{\pm 1/\sqrt{m_X}\}^d$, then we can approximate a linear transformation of the true gradient approximate perturbed by a curvature error as follows

$$\begin{aligned} \mathbf{y} &= \mathcal{A}(X) + \mathbf{e} + \mathbf{z} \\ \mathbf{y}_i &= \frac{1}{\varepsilon}\sum_{j=1}^{m_X} f(x_j + \varepsilon\Phi_{i,j}) - f(x_j) \end{aligned} \tag{10}$$

where $\mathcal{A}(X)_i = \mathrm{tr}(\Phi_i^T X)$ is a linear transformation, $\mathbf{e}$ includes the curvature errors, and z is normally distributed with zero-mean. Notably, the Hermitian adjoint of $\mathcal{A}$ is defined as $A^*(y) = \sum_{j=1}^{m_\phi} \Phi_j y_i$. Then, using the Dantzig selector, we can define a convex optimization problem that recovers a low rank reconstruction of X as follows [4]

$$\min_{M \in \mathbb{R}^{m_X \times d}} ||M||_* \text{ subject to } ||\mathcal{A}^*(y - \mathcal{A}(M))|| \leq \lambda \tag{11}$$

where $||\cdot||_*$ is the nuclear norm, $||\cdot||$ is the spectral norm, and λ regulates the distance of the low-rank reconstruction from the true matrix X in the Frobenius norm.

Using the low-rank reconstruction, we can deduce an approximate transformation $\hat{A}$ to the underlying subspace. To obtain A, we employ Singular Value Decomposition on the recovered matrix

$$X_{DS} = U\Sigma V^T$$

The transformation matrix is given by the k top-most singular vectors so that

$$\hat{A} = U^{(k)}$$

Altogether, the SI phase incurs $m_X(m_\phi + 1)$ queries to the oracle. Since the contexts are sampled uniformly across all arms, the regret in the SI phase is likely to stagnate, inflating the total regret measure R_t.

Using the approximated transformation matrix , we can apply the GP-UCB algorithm on the subspace. In practice,

the implementation of the GP-UCB only deviates from Algorithm 3 by requiring that the algorithm transforms all context vectors to the subspace $z = \hat{A}x$, which marginally affects the computation.

4 Subspace Learning with Sketching: SI-BKB

To investigate the impact of subspace learning on Gaussian Process bandit algorithms, we introduce a novel subspace learning-based algorithm: SI-BKB. The SI-BKB leverages the low-rank reconstruction phase of the SI-BO algorithm and the more computationally efficient Gaussian Process regressor of the BKB algorithm. Remarkably, any contextual bandit algorithm can be transformed into a subspace learning variant by modifying the Bayesian optimization (BO) phase.

Moreover, the SI-BKB algorithm maintains the sub-exponential dependency on the ambient dimension of the upper regret bounds that SI-BO holds. The theoretical proof of the bound for SI-BKB parallels that of SI-BO and is expressed in Theorem 1, and the regret due to GP-UCB is merely replaced by that for SI-BKB. Since both operate on the subspace, they weakly depend on the ambient dimension d, allowing the polynomial term to dominate.

Algorithm 2 The SI-BKB Algorithm

Require: $T, d, m_X, m_\Phi, \lambda, \varepsilon, k$, oracle for f, kernel κ

$C \leftarrow m_X$ context vectors sampled uniformly from actions
for $i \leftarrow 1$ to m_Φ **do**
  $\Phi_i \leftarrow m_X$ samples uniformly from $\{\pm 1/\sqrt{m_\Phi}\}^d$
end for
$\mathbf{y} \leftarrow$ compute using Equation 10 in
$\hat{X}_{DS} \leftarrow$ low-rank reconstruction by the Dantzig Selector
compute the SVD $\hat{X}_{DS} = \hat{U}\Sigma\hat{V}^T$
$\hat{A} \leftarrow \hat{U}^{(k)}$
$\mathcal{D} \leftarrow$ all $(\hat{A}\mathbf{x_t}, y_t)$ pairs queried so far
Apply BKB as specified in Algorithm 1 with transformed context vectors $\hat{x} = \hat{A}x$.

Theorem 1 (Upper Bound on Regret for SI-BKB)**.** *For an oracle with negligible noise during subspace learning, under the same parameter assumptions provided in Theorem 4 in [5] and in Theorem 2 in [3], with probability $1-\delta$, we have that*

$$R_T \;\leq\; \mathcal{O}\big(k^3 d^2 \log^2(1/\delta)\big) + \sqrt{2}\, R_{\mathrm{BKB}}(T,\, g,\, \kappa)$$

where $R_{\mathrm{BKB}}(T,\, g,\, \kappa)$ is the regret bound of the BKB *algorithm applied on the subspace environment reward function g using kernel κ.*

Sketch of Proof. The proof of the regret upper bound mirrors that of Theorem 4 in [5]. The proof sketch presented here is augmented with modifications to derive the upper bounds for SI-BKB. Remarkably, the modifications presented here can be extended to other bandit algorithms in place of BKB.

Assuming, without loss of generality, that the reward is almost surely bounded by 1, from the proof of Lemma 1 in [5], the cumulative regret for SI-BKB is bounded by

$$R_T \leq n + \eta T + R_{\mathrm{BKB}}(T,\, \hat{g},\, \kappa) \tag{12}$$

where $n = m_X(m_\varphi + 1)$ is the duration of the subspace identification, η is the subspace approximation error, and $\hat{g}$ is an approximation of the true reward function g in the learned subspace.

According to Theorem 2 in [3], for a sufficiently large horizon[3],the regret bound on BKB is given by

$$R_{\mathrm{BKB}}(T, \hat{g}, \kappa) = \mathcal{O}\bigg(\sqrt{T}\big(\gamma_T \log C_\kappa T + \log 1/\delta + \sqrt{B_{\hat{g}}\gamma_T \log C_\kappa T}\,\big)\bigg) \tag{13}$$

where A is some constant factor, C_κ is the upper bound on the kernel κ, and $B_{\hat{g}} > ||\hat{g}||_\kappa^2$ is some bound on the complexity of the specified reward function $\hat{g}$, i.e. the squared norm of g concerning the kernel κ in the RKHS κ.

Using the parameter assumptions for Theorem 4 in [5], we have by Lemma 13 in [5] that

$$||\hat{g}||_\kappa^2 \leq 2||g||_\kappa^2 \tag{14}$$

$$\eta = T^{-1/2} \tag{15}$$

Thus, we may select $B_{\hat{g}}$ and B_g so that by Eq. 14

$$||g||_\kappa^2 < B_g \;\text{ and }\; ||\hat{g}||_\kappa^2 < B_{\hat{g}} = 2B_g$$

Hence, observing that Eq. 13 is monotonic increasing with respect to $B_{\hat{g}}$, we have that the regret bound on the approximated reward function is bounded by that of the true reward function as follows

$$R_{\mathrm{BKB}}(T,\, \hat{g},\, \kappa) \leq \sqrt{2} R_{\mathrm{BKB}}(T, g, \kappa) \tag{16}$$

Using Eq. 15, the $\eta T = \sqrt{T}$ term is dominated by the regret bound in Eq. 13, which is reducible to $\mathcal{O}(\sqrt{T}\log T)$. Hence, the ηT term is negligible for the upper bound.

Finally, we bound the regret incurred due to subspace learning. Using the parametrization provided in Theorem 4 in [5], we have that $m_X = \mathcal{O}(kd\log 1/\delta)$ and $m_\varphi = \mathcal{O}(k^2 d \log 1/\delta)$. Thus, the regret incurred by the subspace learning phase is given by

$$\begin{aligned} n &= m_X(m_\varphi + 1) \\ &= \mathcal{O}(k^3 d^2 \log^2 1/\delta) \end{aligned} \tag{17}$$

Applying the bounds derived in Eq. 17 and Eq. 16 to Eq. 12, while treating ηT as negligible, we arrive at the desired upper bound on the cumulative regret.

□

5 Experimental Setup

To evaluate the two algorithms, we use synthetic data to generate contexts in each bandit environment. The implementation of the bandit algorithms and the synthetic data generation is available at https://github.com/Cheese-1/SparseSequentialLearning.

[3] We require that $\log(C_\kappa T) \geq 0$.

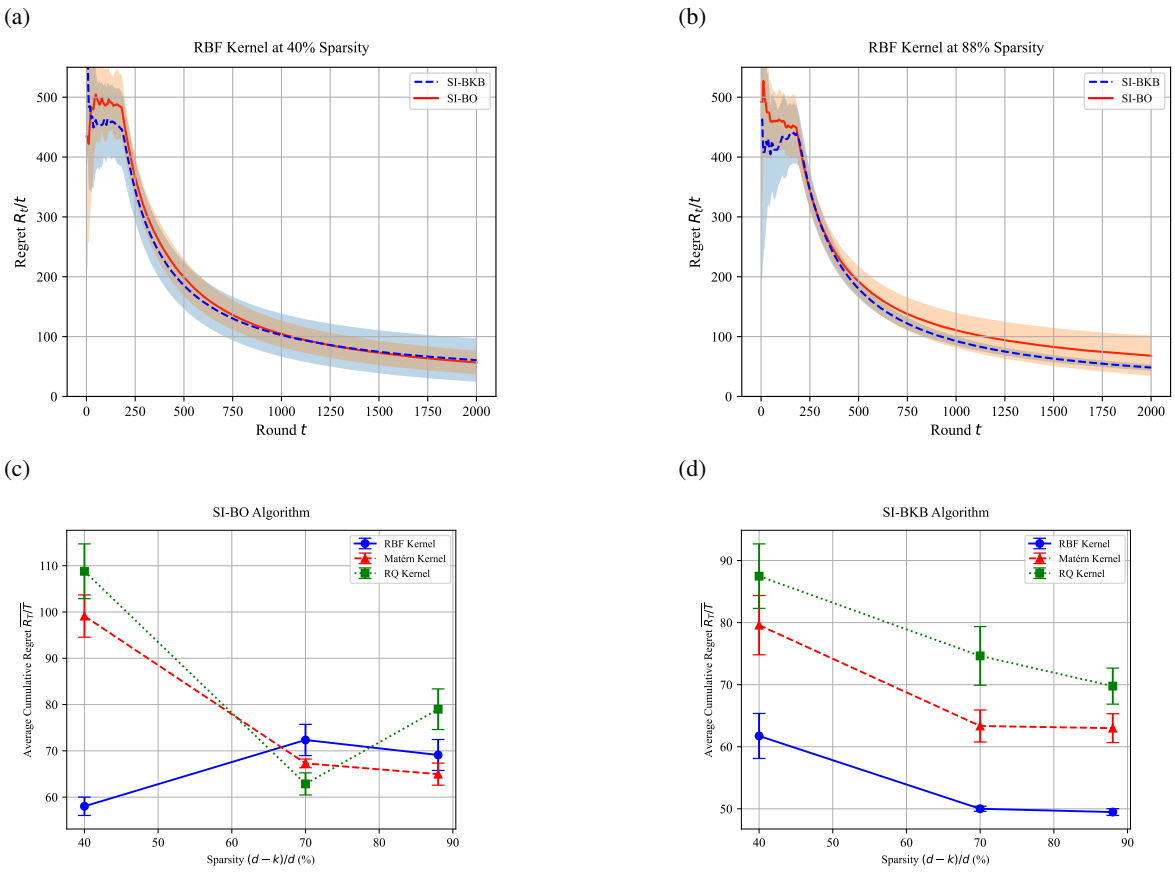

Figure 1: A summary of the regret performances of the bandit algorithms in the linear bandit. Figures a-b depict the regret performance for the RBF kernel in the lowest and highest sparsity conditions (from Fig. 3). Figures c-d depict the average of the cumulative average regret in the final 100 rounds.

5.1 Synthetic Data Generation

Synthetic data provides a controlled and reproducible framework for evaluating bandit algorithms, particularly in structured settings where assumptions such as sparsity or smoothness can be precisely enforced. It is a conventional approach to undertake in evaluating the performance of bandits (*see* [1]).

In this paper, we define a probability distribution $\mathcal{D}_a$ for every action $a \in \mathcal{A}$ over the context space $\mathcal{X} \subseteq \mathbb{R}^d$ to generate synthetic data. In particular, we assume that the distributions are normally distributed $D_a = \mathcal{N}(\xi_a, \sigma_a^2 I_d)$. The mean context ξ_a is fixed and is uniformly sampled from the context space $\mathcal{X}$. The variance is directly specified at $\sigma_a^2 = 0.1^2$ and is fixed across arms.

Since the empirical regret is reliant on the expected reward of the best constant-action algorithm, the environment simulates the reward of each arm to determine the empirical average reward. This simulation is hidden from the bandit algorithm; correspondingly, the exact computation of the regret is also hidden. The simulation is repeated for $10,000$ trials, and the most significant average reward across trials is selected.

Altogether, the synthetic data generation process is highly grounded in probability. To account for the holistic performance of the bandits, each experimental condition is repeated 20 times, and the empirical regret is aggregated across these trials.

5.2 Parameter Configuration

Due to the abundance of modifiable (hyper-)parameters, we separate the (hyper-)parameters into three categories. The first category consists of the environmental variables, which are fixed for each bandit environment. This category includes the reward function f and the variance of the noisy oracle σ^2. The second category comprises parameters that can be computed from the environmental variables. The upper bound on complexity B and the maximal information gain γ_t are among such parameters. The last category consists of parameters that require fine-tuning. These parameters are fine-tuned through grid search by selecting the parameter values that achieve the best regret at the horizon.

The parameter configurations for the `SI-BO` and `SI-BKB` algorithms in each environment are presented in Appendix C.

6 Results

6.1 Linear Bandit Simulation

The linear bandit environment was constructed by embedding a low-dimensional linear function with $k = 3$ in high-

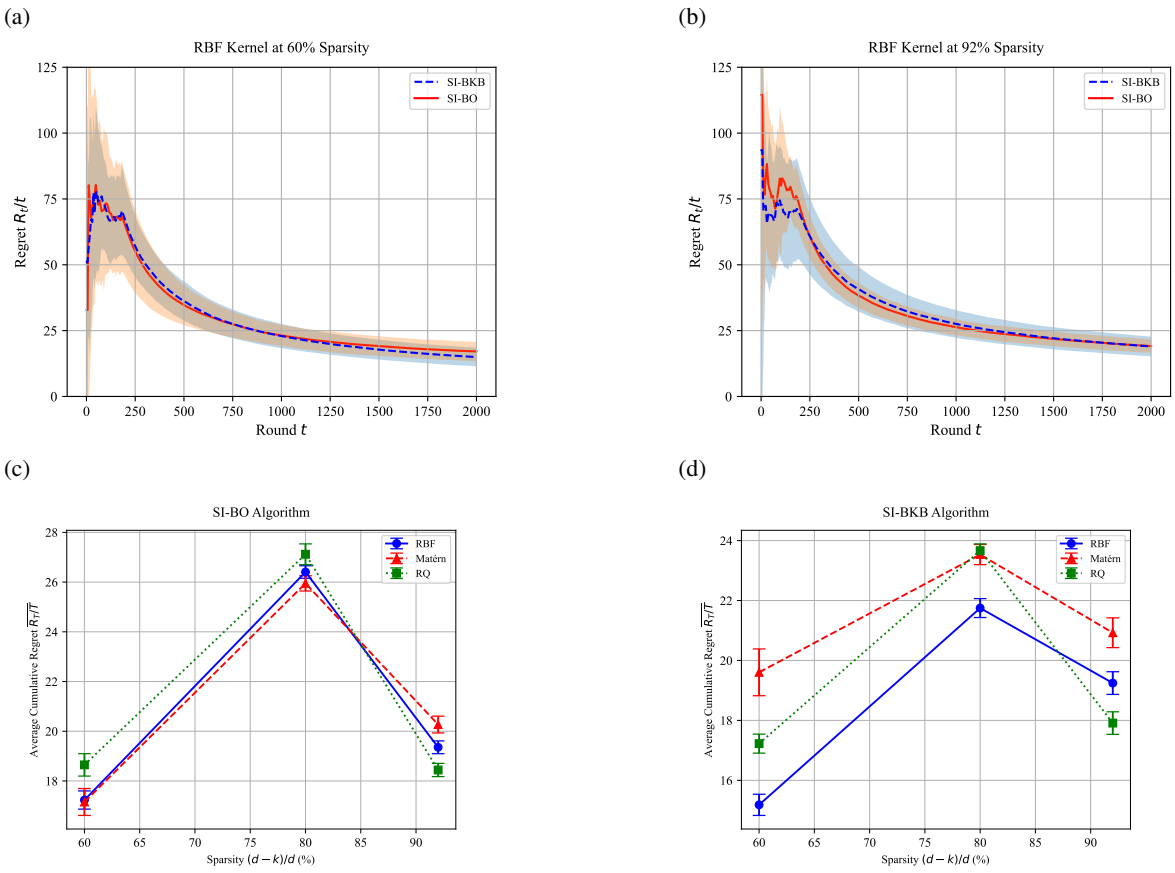

Figure 2: A summary of the regret performances of the bandit algorithms in the Branin bandit. Figures a-b depict the regret performance for the RBF kernel in the lowest and highest sparsity conditions (from Fig. 4). Figures c-d depict the average of the cumulative average regret in the final 100 rounds.

dimensional spaces (see Appendix B). In this experiment, we considered three linear bandit environments with ambient dimensions $d \in \{5, 15, 25\}$. In each case, sparsity was quantified by the percentage of additional dimensions $\frac{d-k}{d}$. In Appendix D, we present in Table 3 the empirical cumulative regret performance for every choice of kernel with tuned parameters under each sparsity condition. Each kernel-sparsity combination was simulated for 20 trials, with errors indicating standard deviation. Fig. 1a-b depicts the select regret performance for the RBF kernel under the lowest and highest sparsity conditions. Fig. 1c-d provides the average of the cumulative average regret in the final 100 rounds. The error is calculated assuming that the covariance between cumulative average regret values is negligible; the convergence of the cumulative average regret supports this assumption.

As observed in Fig. 1a-b, both the `SI-BKB` and `SI-BO` algorithms achieve comparable regret performance. The initial subspace learning phase, whose duration is $m_X(m_\varphi + 1) = 180$, is depicted with an erratic and high regret. Both algorithms also demonstrate an initial *cold-start* following the SI phase, lasting up to round $t \approx 250$. This *cold-start* is likely due to initial high exploration by the BO phase algorithm, as well as the high regret cost incurred by the SI phase. Finally, both algorithms exhibit a strong decay to a low regret value. Notably, the average performance of the `SI-BKB` algorithm consistently lies within one standard deviation of that of `SI-BO` during the decay, indicating similar behavior and action selection.

The choice of kernel also appears to influence the final regret under certain conditions.. The RBF kernel tends to approximate the reward function with less variance in low-sparsity conditions, commensurate with its capability to approximate linear functions [11]. Under high sparsity conditions, the performance of the RQ and Matérn kernels visibly improves.

6.2 Non-linear Bandit Simulation: Branin function

The linear bandit environment was constructed by embedding the low-dimensional Branin function with $k = 2$ in high-dimensional spaces (see Appendix B). Unlike the linear bandit, the Branin function introduces additional curvature. In this experiment, we considered three linear bandit environments with ambient dimensions $d \in \{5, 15, 25\}$. In each case, sparsity was quantified by the percentage of additional dimensions $\frac{d-k}{d}$. In Appendix D, we present the empirical cumulative regret performance for every choice of kernel with tuned parameters under each sparsity condition in Ta-

ble 4. Each kernel-sparsity combination was simulated for 20 trials, with errors indicating standard deviation. Fig. 2a-b depicts the select regret performance for the RBF kernel under the lowest and highest sparsity conditions. Fig. 2c-d provides the average of the cumulative average regret in the final 100 rounds. The error is calculated assuming that the covariance between cumulative average regret values is negligible; the convergence of the cumulative average regret supports this assumption.

Similar to the linear bandit, as observed in Fig. 2a-b, both the SI-BKB and SI-BO algorithms achieve comparable regret performance. The initial subspace learning phase, whose duration is $m_X(m_\varphi + 1) = 180$, is depicted with an erratic and high regret. Both algorithms also demonstrate an initial *cold-start* following the SI phase, lasting up to round $t \approx 200$. This *cold-start* is likely due to initial high exploration by the BO phase algorithm, as well as the high regret cost incurred by the SI phase. Finally, both algorithms exhibit a strong decay to a low regret value. Notably, the average performance of the SI-BKB algorithm consistently lies within one standard deviation of that of SI-BO during the decay, indicating similar behavior and action selection.

Across kernels, the regret performance does not appear to deviate significantly. Under the same sparsity condition, the final regret performance does not considerably deviate across kernels, suggesting that any of the three kernels may equally approximate the reward function. Nonetheless, we observe an anomaly at $d = 15$ for both algorithms, where the regret spikes.

7 Discussion

In both linear and non-linear environments, the regret performance is inconsistent with the theoretically desirable no regret property. As observed in Fig. 1a-b and Fig. 2a-b, the cumulative average regret plateaus at a non-zero value, indicating that the bandit algorithms consistently select suboptimal arms. Moreover, this discrepancy is exhibited equally by both SI-BO and SI-BKB. This observation suggests that the convergence to suboptimal arms is likely due to the misalignment of the learned subspace.

A possible mechanism by which misalignment may affect regret may originate from irrelevant dimensions in the context vector. An irrelevant context vector x_i is an axis of the context vector that is invariable relative to the reward function, so that $\frac{\partial f}{\partial x_i} = 0$. When the transformed context $\hat{A}x$ depends on the value of irrelevant context dimensions, any approximated reward function $\hat{g}$ is also likely to depend on them by proxy. When the subspace dimension is correctly specified, the irrelevant context dimensions serve as noise in the transformed context vector. Correspondingly, This misalignment ultimately mutates the objective of the bandit algorithm in the BO phase by forcing the approximated reward function to account for noisy input.

Another surprising observation is that the regret performance of the two bandit algorithm does not necessarily deteriorates under high sparsity. In fact, as shown in Fig. 1d, the regret performance of SI-BKB in the linear bandit substantially improves under harsher sparse constraint. Fig. 1c also demonstrates this improvement for the SI-BO algorithm but only for the Matérn and RQ kernels. The latter observation possibly suggests that high sparsity, through misalignment, relaxes the constraints of the approximations, allowing relatively unsuitable kernel choice to produce better approximations. Notably, this relaxation does not always occur evident by the regret spike at 80% sparsity in the Branin bandit (*see* Fig. 2c-d).

7.1 Limitations

A major limitation to the practical application of the SI-BO and SI-BKB algorithm is the underlying stringent assumptions necessary to ensure their validity. The occurrence of a smooth reward function that may be invoked through an oracle anytime with an appreciable sensitivity to small perturbations in the context vector is rare in practice. Moreover, SI-BO and SI-BKB require prior knowledge several environmental variables—such as oracle noise σ, function complexity upper bound B, maximal information gain γ_t. While existing literature provides frameworks to handle the latter (*see* the doubling trick [7]), this regiment of stringent requirements discourages further exploration the subspace learning framework as-is.

Another limitation arises from the high computational cost incurred by the SI-BO and SI-BKB. In both scenarios, the index function is computed in part through multiplication of matrices which scale with the number of oracle queries. Without a mechanism to ignore past oracle queries, for large horizons, the space complexity alone renders computation an impossibility.

A methodological limitation to the analysis presented in this paper lies in the generation of synthetic data. Per arm, since the contexts are normally distributed, the generated context vectors lie in close proximity in the learned subspace. This approach to synthetic generation precludes environments in which arm distributions $\mathcal{D}_a$ are more expressive.

8 Conclusion

This paper aimed to investigate the regret performance of bandit algorithms incorporating Subspace Learning and Gaussian Processes under sparsity restrictions. To that end, this paper implemented an adaptation of the SI-BO algorithm and a novel algorithm SI-BKB that improves computational efficiency. Moreover, this paper demonstrates that SI-BKB achieves sub-exponential upper bound on regret with respect to the ambient dimension similarly to SI-BO. Empirically, SI-BKB achieved comparable regret performance to SI-BO in both linear and non-linear environments. Additionally, this paper demonstrates empirically that both algorithm are comparably performant under high sparsity conditions.

While the SI-BKB algorithm enhances the computational efficiency of the Bayesian Optimization phase, the choice of a Gaussian Process UCB-based algorithm is mostly unconstrained and independent of the Subspace Learning framework. By attaching the Subspace Identification phase, bandit algorithms may be adapted to sparse setting, possibly limiting the regret bound to be sub-exponential with respect to the

ambient dimension. The practical utility of such algorithms compared to existing sparse algorithms is yet to be explored in current literature.

Likewise, the separation of the two phases of the SI-BO and SI-BKB algorithms, while suitable for stationary environments, may be further generalized. With sufficient modification, the subspace learning may be run simultaneously with some bandit algorithm, reducing the high initial regret cost. Alternatively, the learned subspace may be refined at later rounds given the abundance of oracle queries.

9 Responsible Research

To ensure the reproducibility of the paper, this paper endeavored to outline all aspects of the empirical experiments. Sections 3-4 outlined the mathematics used to implement all tested algorithms. The methodology for the synthetic data generation is summarized in Section 5. The parametrization employed is provided Appendix B. All the generated data is viewable in Appendix D. Finally, the entire empirical study pipeline is implemented in a public repository, which is available through this paper. Altogether, all aspects of the empirical study are provided, allowing other researchers to verify the results exhibited here independently.

While this paper does not directly address real-world applications, we acknowledge the potential for misuse of contextual bandit algorithms. In the domains of personalized advertising, dynamic pricing, or automated decision support, malicious actors may deploy contextual bandits to facilitate predatory and unfair practices. Hence, the application of contextual bandit algorithms must proactively ground its development in mitigating, if not eliminating, such misuse.

References

[1] Baruch Awerbuch and Robert Kleinberg. Online linear optimization and adaptive routing. *Journal of Computer and System Sciences*, 74(1):97–114, February 2008.

[2] Hamsa Bastani and Mohsen Bayati. Online Decision Making with High-Dimensional Covariates. *Operations Research*, 68(1):276–294, January 2020. Publisher: INFORMS.

[3] Daniele Calandriello, Luigi Carratino, Alessandro Lazaric, Michal Valko, and Lorenzo Rosasco. Gaussian process optimization with adaptive sketching: Scalable and no regret, 2019.

[4] Emmanuel Candès and Yaniv Plan. Tight oracle inequalities for low-rank matrix recovery from a minimal number of noisy random measurements. *Information Theory, IEEE Transactions on*, 57:2342 – 2359, 05 2011.

[5] Josip Djolonga, Andreas Krause, and Volkan Cevher. High-Dimensional Gaussian Process Bandits. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

[6] Tyagi Hemant and Volkan Cevher. Active Learning of Multi-Index Function Models. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[7] Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, 1 edition, July 2020.

[8] 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*, pages 661–670, April 2010. arXiv:1003.0146 [cs].

[9] Vianney Perchet and Philippe Rigollet. The Multi-Armed Bandit Problem with Covariates. *The Annals of Statistics*, 41(2):693–721, 2013. Publisher: Institute of Mathematical Statistics.

[10] Joaquin Quiñonero-Candela, Carl Edward Rasmussen, and Christopher K. I. Williams. Approximation Methods for Gaussian Process Regression. In Léon Bottou, Olivier Chapelle, Dennis DeCoste, and Jason Weston, editors, *Large-Scale Kernel Machines*, pages 203–224. The MIT Press, August 2007.

[11] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 2006.

[12] Philippe Rigollet and Assaf Zeevi. Nonparametric Bandits with Covariates, March 2010. arXiv:1003.1630 [math].

[13] Bernhard Schölkopf. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. Adaptive computation and machine learning. MIT Press, Cambridge, Mass, 2002.

[14] Weiwei Shen, Jun Wang, Yu-Gang Jiang, and Hongyuan Zha. Portfolio choices with orthogonal bandit learning. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, page 974–980. AAAI Press, 2015.

[15] Aleks Slivkins. Contextual Bandits with Similarity Information. *Journal of Machine Learning Research*, 15(73):2533–2568, 2014.

[16] Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias W. Seeger. Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, 58(5):3250–3265, 2012.

A Notation Table

Table 1: The notation table for important notation used in the paper. The definition of each symbol is provided.

Symbol	Definition
$\mathcal{A}$	collection of arms, or actions
d	dimension of the ambient space
$\mathcal{X} \subseteq \mathbb{R}^d$	set of context vectors
$\mathcal{D}_a$	context vector distribution for arm a
T	the horizon
$x_{t,a}$	context vector sampled at round t for arm a
y_t	reward returned by the oracle
$\mathcal{D}$	set of oracle queries
σ^2	variance of the oracle's noise
η_t	sampled oracle noise $\eta_t \sim \mathcal{N}(0, \sigma^2)$
$f : \mathbb{R}^d \to \mathbb{R}$	reward function over the ambient space
k	dimension of the effective subspace
A	transformation matrix from $\mathbb{R}^d$ to $\mathbb{R}^k$
$g : \mathbb{R}^k \to \mathbb{R}$	the true reward function over the subspace
$\hat{A}$	the reconstructed transformation matrix
$\hat{g} : \mathbb{R}^k \to \mathbb{R}$	the approximated g over the learned subspace
X_t	the reward random variable at round t
R_T	the cumulative reward up to round T
$\mathrm{UCB}_t(\cdot)$	the UCB index function
X_t^*	matrix of previously observed contexts
y	vector of previously observed rewards
$\kappa(\cdot, \cdot)$	the RKHS kernel for the Gaussian Process
$k_t(x)$	covariance vector of x with contexts in X_t^*
$K(X_t^*)$	covariance matrix for the contexts in X_t^*
δ	accuracy rate
γ_t	maximal information gain at round t
$\|\| \cdot \|\|_\kappa$	norm induced by the kernel κ
B, B_g, $B_{\hat{g}}$	upper bound for function complexity $\|\|g\|\|_\kappa^2$
β_t	exploration factor
$\mu_t(\cdot), \sigma_t(\cdot)$	posterior mean and standard deviation
ℓ	(global) length-scale for the kernel
ν	smoothness characteristic for the Matérn kernel
α	variance weight for the RQ kernel
$S_t \subseteq \mathbb{R}^d$	set of inducing points
m_t	size of S_t
$z_t(\cdot)$	the Nyström embedding for contexts
K_{S_t}	covariance matrix for the inducing points
$k_{S_t}(x)$	covariance vector of x with contexts in S_t
$Z_t(X_t)$	matrix of the Nyström embeddings of X_t
V_t	an $m_t \times m_t$ invertible matrix
$\tilde{\mu}_t, \tilde{\sigma}_t$	approximated mean and uncertainty for BKB
$\bar{q}$	scaling factor for inclusion probability
m_X	number of context samples
m_φ	number of direction samples per context
X	the gradient matrix for the reward over samples
Φ_i	direction vector matrix
$\mathcal{A}(\cdot)$	linear transformation using direction vectors
ε	step length
$\mathbf{y}$	sample space vector
$\|\| \cdot \|\|_*$	nuclear norm
$\|\|M\|\|$	spectral norm for matrix M
λ	acceptable distance for the reconstructed matrix
X_{DS}	the low-rank reconstructed matrix
$\hat{U}^{(k)}$	k top-most singular vector of $\hat{U}$
C_κ	bound on the kernel, i.e. $\kappa(x, x') \leq C_\kappa$

B Additional Algorithms

This appendix contains the outline for the GP-UCB and SI-BO algorithms. These algorithms are not directly related to the main contribution—the SI-BKB algorithm—and are therefore provided here for reference.

Algorithm 3 The GP-UCB Algorithm adapted from [16]

Require: kernel function κ, B, δ, σ, horizon T

Initialize the set of oracle queries $\mathcal{D} = \emptyset$
for $t = 1$ to T **do**
  Observe the context $x_{t,a}$ for every action $a \in \mathcal{A}$.
  Compute the posterior mean $\mu_t(x)$ and variance $\sigma_t(x)$ for every context vector using Eq. 5.
  Compute the factor $\beta_t = 2B + 300\gamma_t \log^3(t/\delta)$
  Select $a_t = \arg\max_{a \in \mathcal{A}} \mu_t(x_{t,a}) + \beta_t^{1/2} \sigma_t(x_{t,a})$
  Query the oracle in Eq. 1 to observe y_t
  $\mathcal{D} \leftarrow \mathcal{D} \cup \{(x_{a_t,t}, y_t)\}$
end for

Algorithm 4 The SI-BO algorithm adapted from [5]

Require: T, d, m_X, m_Φ, λ, ε, k, oracle for f, kernel κ

$C \leftarrow m_X$ context vectors sampled uniformly from actions
for $i \leftarrow 1$ to m_Φ **do**
  $\Phi_i \leftarrow m_X$ samples uniformly from $\{\pm 1/\sqrt{m_\Phi}\}^d$
end for
$\mathbf{y} \leftarrow$ compute using Equation 10
$\hat{X}_{DS} \leftarrow$ low-rank reconstruction by the Dantzig Selector
compute the SVD $\hat{X}_{DS} = \hat{U}\Sigma\hat{V}^T$
$\hat{A} \leftarrow \hat{U}^{(k)}$
$\mathcal{D} \leftarrow$ all $(\hat{A}\mathbf{x_t}, y_t)$ pairs queried so far
Apply GP-UCB as specified in Algorithm 3 with transformed context vectors $z = \hat{A}x$.

C Fine-tuned Parametrization

Table 2: Parameters for the Linear Bandit for the SI-BO algorithm.

Parameter	Value	Explored Range
T	2000	N.A.
d	N.A.	$5, 15, 25$
k	3	N.A.
$\mathcal{X}$	$[-10, 10]^d$	N.A.
$f(x)$	$32x_1 - 16x_4 + 8x_0 - 45$	N.A.
m_X	30	N.A.
m_φ	5	N.A.
B	1000	N.A.
γ_t	$\frac{1}{2}\log\det(I_t + K_t(X_t^*)/\sigma^2)$	N.A.
δ	0.95	N.A.
σ	0.1	N.A.
ε	0.001	N.A.
λ	computed in [6]	N.A.
$\ell_{\text{Matérn}}$	12.5	$10, 12.5, 15, 17.5$
ν	2.5	$0.5, 1.5, 2.5$
ℓ_{RBF}	12.5	$10, 12.5, 15, 17.5$
ℓ_{RQ}	10	$10, 12.5, 15, 17.5$
α_{RQ}	2	$0.5, 1, 2$

Table 3: Parameters for the Linear Bandit for the SI-BKB algorithm.

Parameter	Value	Explored Range
T	2000	N.A.
d	N.A.	$5, 15, 25$
k	3	N.A.
$\mathcal{X}$	$[-10, 10]^d$	N.A.
$f(x)$	$32x_1 - 16x_4 + 8x_0 - 45$	N.A.
m_X	30	N.A.
m_φ	5	N.A.
B	1000	N.A.
γ_t	$\sum_{i=1}^t \tilde{\sigma}_t(\tilde{x}_i)$	N.A.
δ	0.95	N.A.
σ	0.1	N.A.
α	$\frac{1+0.7}{1-0.7}$	N.A.
C_κ	50	N.A.
$\bar{q}$	10	[7.5, 10, 12.5]
ε	0.001	N.A.
λ	computed in [6]	N.A.
$\ell_{\text{Matérn}}$	12.5	$10, 12.5, 15, 17.5$
ν	2.5	$0.5, 1.5, 2.5$
ℓ_{RBF}	12.5	$10, 12.5, 15, 17.5$
ℓ_{RQ}	10	$10, 12.5, 15, 17.5$
α_{RQ}	2	$0.5, 1, 2$

Table 4: Parameters for the Branin Bandit for the `SI-BO` algorithm.

Parameter	**Value**	**Explored Range**
T	2000	N.A.
d	N.A.	$5, 15, 25$
k	2	N.A.
$\mathcal{X}$	$[-5, 15]^d$	N.A.
$f(x)$	$-\Big(\frac{x_1+x_3}{\sqrt{2}} - \frac{5.1}{4\pi^2}\big(\frac{x_0+x_2}{\sqrt{2}}\big)^2 + \frac{5(x_0+x_2)}{\pi\sqrt{2}} - 6\Big)^2 - 10\Big(1 - \frac{8}{\pi}\cos\big(\frac{x_0+x_2}{\sqrt{2}}\big)\Big) - 10$	N.A.
m_X	30	N.A.
m_φ	5	N.A.
B	1000	N.A.
γ_t	$\frac{1}{2}\log\det(I_t + K_t(X_t^*)/\sigma^2)$	N.A.
δ	0.95	N.A.
σ	0.1	N.A.
ε	0.001	N.A.
λ	computed in [6]	N.A.
$\ell_{\text{Matérn}}$	15	$10, 12.5, 15, 17.5$
ν	2.5	$0.5, 1.5, 2.5$
ℓ_{RBF}	17.5	$10, 12.5, 15, 17.5$
ℓ_{RQ}	17.5	$10, 12.5, 15, 17.5$
α_{RQ}	1	$0.5, 1, 2$

Table 5: Parameters for the Branin Bandit for the SI-BKB algorithm.

Parameter	Value	Explored Range
T	2000	N.A.
d	N.A.	$5, 15, 25$
k	2	N.A.
$\mathcal{X}$	$[-5, 15]^d$	N.A.
$f(x)$	$-\Bigg(\frac{x_1+x_3}{\sqrt{2}} - \frac{5.1}{4\pi^2}\big(\frac{x_0+x_2}{\sqrt{2}}\big)^2 + \frac{5(x_0+x_2)}{\pi\sqrt{2}} - 6\Bigg)^2 - 10\Bigg(1 - \frac{8}{\pi}\cos\big(\frac{x_0+x_2}{\sqrt{2}}\big)\Bigg) - 10$	N.A.
m_X	30	N.A.
m_φ	5	N.A.
B	1000	N.A.
γ_t	$\frac{1}{2}\log\det(I_t + K_t(X_t^*)/\sigma^2)$	N.A.
δ	0.95	N.A.
σ	0.1	N.A.
α	$\frac{1+0.7}{1-0.7}$	N.A.
C_κ	50	N.A.
$\bar{q}$	10	[7.5, 10, 12.5]
ε	0.001	N.A.
λ	computed in [6]	N.A.
$\ell_{\text{Matérn}}$	15	$10, 12.5, 15, 17.5$
ν	2.5	$0.5, 1.5, 2.5$
ℓ_{RBF}	17.5	$10, 12.5, 15, 17.5$
ℓ_{RQ}	17.5	$10, 12.5, 15, 17.5$
α_{RQ}	1	$0.5, 1, 2$

D Empirical Results

The complete empirical results are presented in this appendix. The data for the two experiments was generated by the procedure outlined in Section 5 with the parameters stated in Appendix B. The data for the linear bandit is presented in Figure 3. For the non-linear Branin bandit, the data is presented in Figure 4.

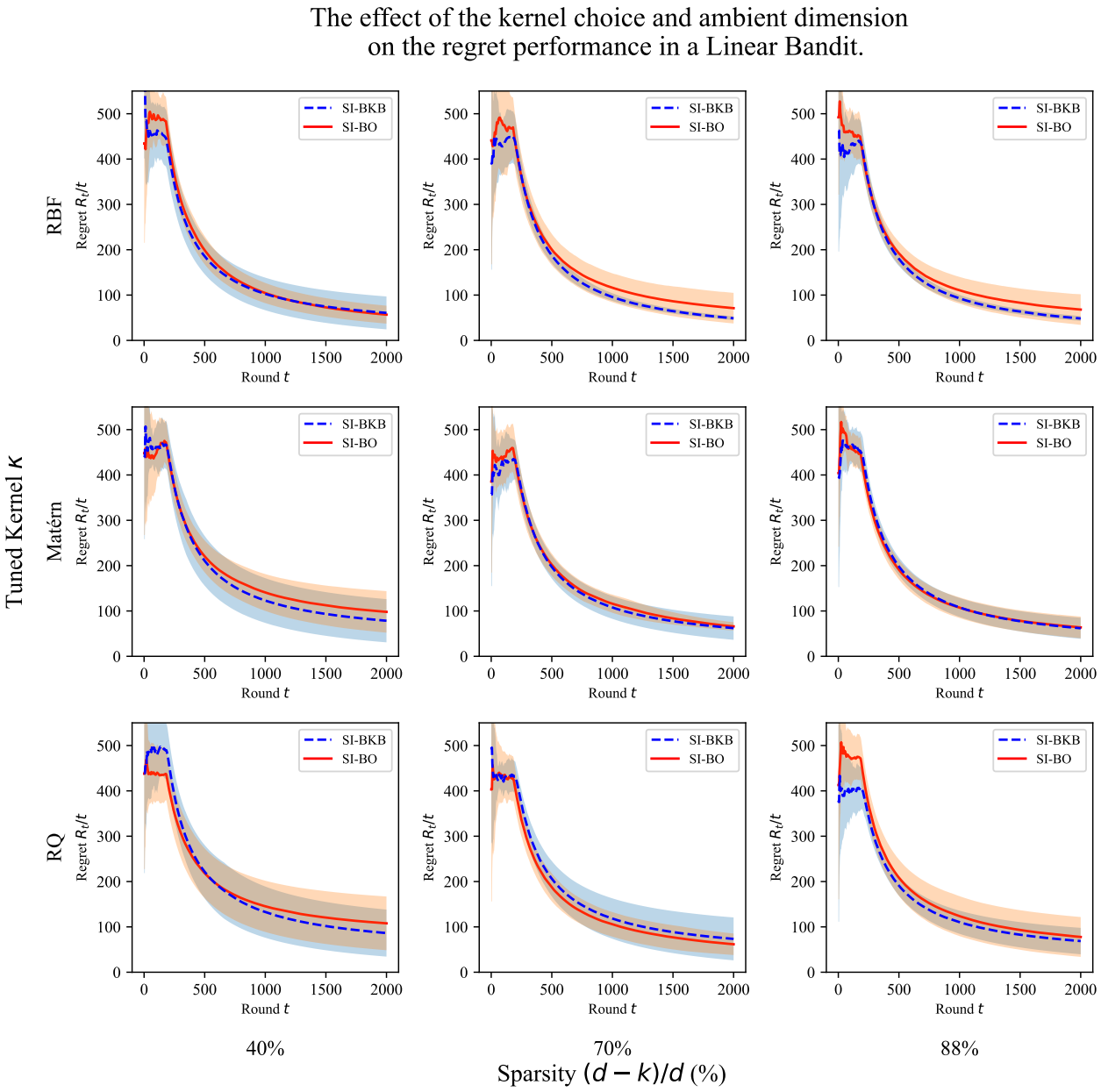

Figure 3: The performance of the SI-BO and SI-BKB algorithms across varying sparsity conditions and kernel choices. The algorithms have a fixed duration for the SI phase and have already been fine-tuned.

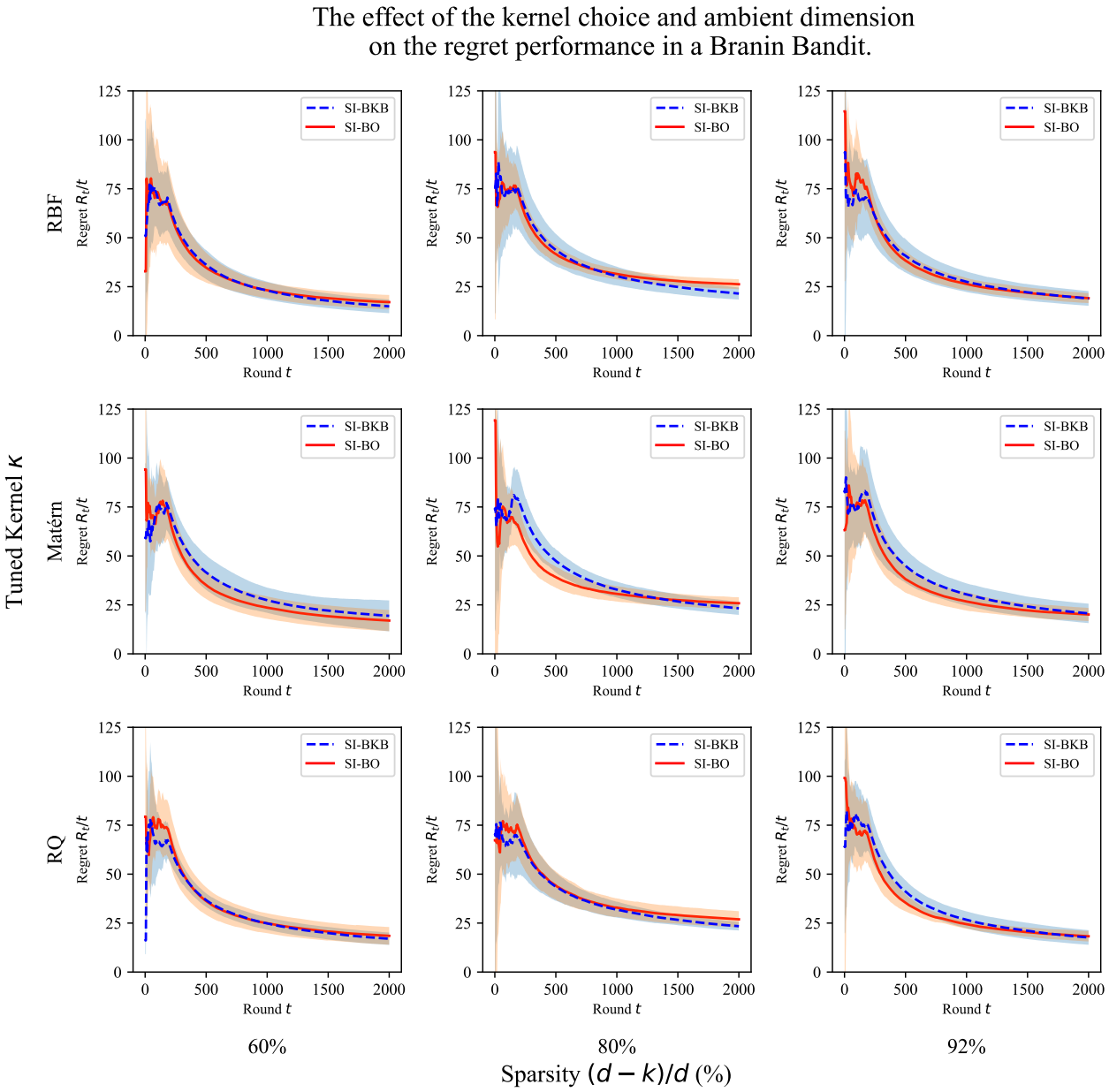

Figure 4: The performance of the SI-BO and SI-BKB algorithms across varying sparsity conditions and kernel choices. The algorithms have a fixed duration for the SI phase and have already been fine-tuned.